



PATENT ABSTRACTS OF JAPAN

(11) Publication number: **06019592 A**(43) Date of publication of application: **28 . 01 . 94**

(51) Int. Cl.

G06F 1/32
G06F 11/34(21) Application number: **04170624**(71) Applicant: **RICOH CO LTD**(22) Date of filing: **29 . 06 . 92**(72) Inventor: **KASATANI KIYOSHI****(54) POWER SAVING CONTROL SYSTEM FOR
COMPUTER USING EQUIPMENT**

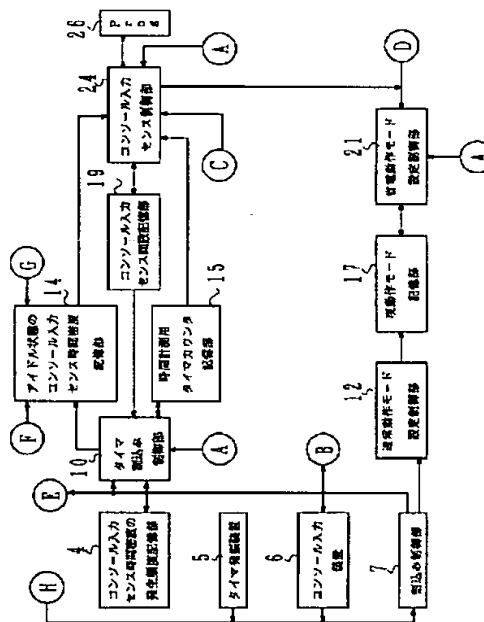
analyzes the generating frequency.

COPYRIGHT: (C)1994,JPO&Japio

(57) Abstract:

PURPOSE: To attain the power saving of an equipment using a computer by stepwise setting each device including a CPU or a display device or the like which are always used in a power saving operation mode at an arbitrarily set time interval in the order of the device whose restoring time is shorter.

CONSTITUTION: Each time a console input sense request is issued from a program 26, a console input sense control part 24 stores the frequency in a console input sense frequency storage part 19. And also, a transmission at a constant period is generated by a timer transmitter 5 in order to measure a prescribed time, it is transmitted through an interruption control part 7, an interruption is generated in a CPU, and a timer interruption control part 10 is started. The timer interruption control part 10 obtains the time density of the console input sense frequency stored in the console input sense frequency storage part 19 by using the counter value of a time measuring use timer counter storage part 15. Moreover, the timer interruption control part 10 measures the generating frequency, stores it in a generating frequency storage part 4, and



(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-19592

(43)公開日 平成6年(1994)1月28日

(51)Int.Cl.⁵

G 0 6 F 1/32
11/34

識別記号

庁内整理番号

F I

技術表示箇所

M 9290-5B
7165-5B

G 0 6 F 1/ 00

3 3 2 A

審査請求 未請求 請求項の数5(全 31 頁)

(21)出願番号 特願平4-170624

(22)出願日 平成4年(1992)6月29日

(71)出願人 000006747

株式会社リコー

東京都大田区中馬込1丁目3番6号

(72)発明者 笠谷 潔

東京都大田区中馬込1丁目3番6号 株式
会社リコー内

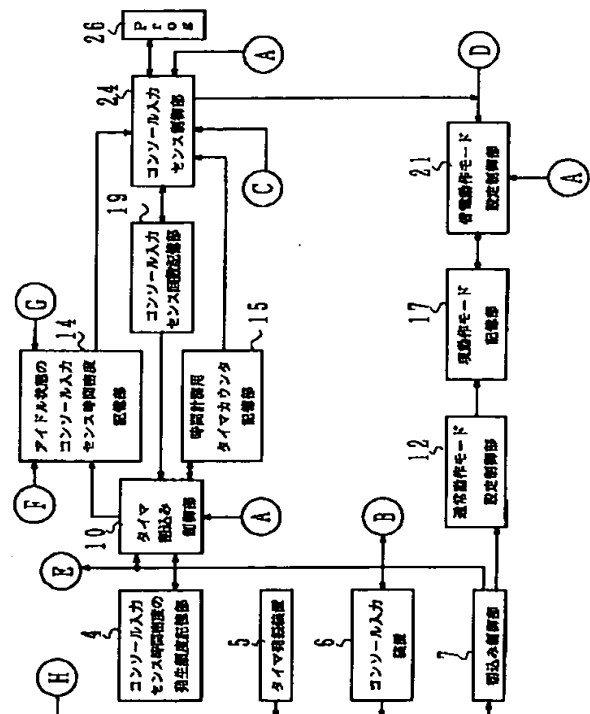
(74)代理人 弁理士 磯村 雅俊

(54)【発明の名称】 コンピュータ利用機器の省電制御システム

(57)【要約】

【目的】 標準入力装置に対する入力センスを行なう各々のプログラムに対応して、コンピュータを利用した機器の省電化を効率良く行なう。

【構成】 割込み制御時に、プログラムの要求による入力センス回数の単位時間の値別の発生頻度に基づき、入力センスのみを行なうアイドル状態を特定するアイドル状態特定部と、プログラムの要求に対応する処理を行なうと共に、入力センス回数の単位時間の値を測定して、アイドル状態の値と照合して、アイドル状態を検出するアイドル状態検出部とを設け、このアイドル状態検出部によるアイドル状態の検出に基づき、通常の動作モードから省電動作モードへの切り替えを行なう省電制御システム。



【特許請求の範囲】

【請求項1】 コンピュータを利用した機器の省電制御システムであり、システムを構成するそれぞれの回路を、通常の動作モードから、より消費電力の低い省電動作モードに切替え、システムの動作電力の低減をはかるコンピュータ利用機器の省電制御システムにおいて、プログラムから、所定の標準入力装置における入力データの発生の有無の問い合わせが要求された回数であるセン

10 ス回数を計数して記憶するセンス回数記憶手段と、プログラムからの上記標準入力装置の制御以外の要求、もしくは、定常的に繰返し発生する割込み以外の上記標準入力装置を含む周辺装置からの割込み発生時に、上記セン

20 ス回数記憶手段で記憶している上記センス回数をゼロ回にリセットするセンス回数リセット手段と、所定の時間間隔で繰り返される割込み時に、上記センス回数記憶手段で記憶している上記センス回数を読み取り、該センス回数の所定の単位時間あたりの回数であるセンス時間密度を測定して記憶するセンス時間密度記憶手段と、該セ

30 ス時間密度記憶手段で記憶しているセンス時間密度の、それぞれの値の所定の単位時間内における発生頻度を測定し、該発生頻度の測定結果で、所定の発生頻度値を超えているセンス時間密度を、プログラムが上記標準入力装置の入力データ発生を待ち続けているアイドル状態を示すアイドル状態センス時間密度として抽出して記憶するアイドル状態特定手段と、上記センス時間密度記憶手段で記憶している上記センス時間密度と、上記アイ

40 ドル状態特定手段で記憶しているアイドル状態センス時間密度との一致を照合して、アイドル状態の検出を行なうアイドル状態検出手段と、該アイドル状態検出手段によるアイドル状態の検出に基づき、上記通常の動作モードから省電動作モードへの切り替えを行なう省電動作モード設定制御手段と、上記省電動作モード中における、プログラムからの上記標準入力装置の制御以外の要求、もしくは、定常的に繰返し発生する割込み以外の上記標準入力装置を含む周辺装置からの割込み発生時に、上記省電動作モードを、通常の動作モードへの復帰させる通常動作モード設定制御手段とを設けることを特徴とするコンピュータ利用機器の省電制御システム。

【請求項2】 請求項1に記載のコンピュータ利用機器の省電制御システムにおいて、プログラムから、所定の標準入力装置における入力データの入力

40 が要求された時に該入力データが存在しない場合、該入力データが発生するまでを、上記省電動作モード設定手段を用いて、省電動作モードに切り替える標準入力制御手段を設けることを特徴とするコンピュータ利用機器の省電制御システム。

【請求項3】 請求項1、もしくは、請求項2のいずれかに記載のコンピュータ利用機器の省電制御システムにおいて、現在実行中のプログラムからのプログラム実行終了要求に基づき、上記アイドル状態特定手段で記憶し

50

ているアイドル状態センス時間密度を、終了を要求したプログラムに付与して終了させるプログラム終了制御手段を設けることを特徴とするコンピュータ利用機器の省電制御システム。

【請求項4】 請求項1から請求項3のいずれかに記載のコンピュータ利用機器の省電制御システムにおいて、上記省電動作モード設定制御手段を用いて、省電動作モードに切り替える時、システムを構成するそれぞれの回路を、所定の順序、および、所定の時間間隔で、省電動作モードに切り替える省電動作モードフェーズ制御手段と、上記通常動作モード設定制御手段を用いて、省電動作モード中から通常の動作モードへ復帰する時、システムを構成するそれぞれの回路を、所定の順序、および、所定の時間間隔で、通常の動作モードに復帰させる通常動作モードフェーズ制御手段とを設けることを特徴とするコンピュータ利用機器の省電制御システム。

【請求項5】 請求項1から請求項4のいずれかに記載のコンピュータ利用機器の省電制御システムにおいて、実行対象の任意のプログラムに対応して、少なくとも、上記センス時間密度記憶手段で用いる上記センス時間密度の測定用の所定の単位時間と、上記アイドル状態特定手段で用いる、上記センス時間密度の発生頻度の測定用の所定の単位時間、および、アイドル状態センス時間密度の抽出用の所定の発生頻度値と、上記省電動作モードフェーズ制御手段で用いる、所定の順序、および、所定の時間間隔と、上記通常動作モードフェーズ制御手段で用いる、所定の順序、および、所定の時間間隔とを含む省電制御条件の設定値を変更する省電制御条件設定制御手段を設けることを特徴とするコンピュータ利用機器の省電制御システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、パーソナルコンピュータやワークステーションなどのコンピュータ利用機器の省電化（省電力化）技術に係わり、特に、電力供給の停止によるスループットの低下を回避して、効率の良い省電化を行なうのに好適なコンピュータ利用機器の省電制御システムに関するものである。

【0002】

【従来の技術】 現在、ワークステーションやパーソナルコンピュータなど、コンピュータ技術を利用した機器の小型化が進められている。特に、携帯型の機器では、携帯性を向上させるために、電池での動作時間を伸ばす必要があり、電池の供給能力を高めたり、例えば、「日経エレクトロニクス 1990 4-2 (no. 496)」(1990年、日経BP社発行)のpp95~120に記載のように、機器の消費電力を下げる技術が検討されている。しかし、電池の供給能力を高めるには、電池が大きくなり、小型化に限界がある。

【0003】 機器の消費電力を下げる技術では、特に、

機器を構成する装置で未使用な装置に供給している無駄な電力を供給しないように制御するものがある。しかし、コンピュータにおいては、どの装置が未使用であるのかを、自動的に判別するのは困難である。なぜなら、コンピュータは、通常、プログラムを外部記憶媒体からメモリに読み込んで実行する、いわゆるノイマン型が主流であり、様々なプログラムを動作させることができる。このため、コンピュータが、動作させるプログラムの内容を予め十分に知っていなければ、どの装置が使用されるかを特定できない。しかし、コンピュータが、全てのプログラムの内容を知ることが不可能である。

【0004】このような問題を解決するために、従来は、例えば、ある装置に対してプログラムからのアクセスが一定時間ない時に、その装置に対する電力の供給を停止し、その後、その装置へのアクセスを監視し、アクセスが発生すれば電力の供給を再開していた。この技術では、電力が供給されていない装置に対するアクセスが発生した場合には、その電力供給を再開して、その装置の準備が整うまで、そのアクセスを一時的に待たせておく。

【0005】装置の準備には時間がかかるため、電力供給の停止後のアクセスが頻繁に発生すると、プログラムのスループットを大きく損なう恐れがある。そのため、電力供給の停止の基準となる「一定時間」を、あまり短くすることができない。また、この「一定時間」を、あまり長くすると、省電効果が、あまり期待できなくなってしまう。このようなことから、「一定時間」の設定は、通常、ユーザに任されている。

【0006】

【発明が解決しようとする課題】解決しようとする問題点は、従来技術では、使用されるプログラムの種類に係わらず、電力供給の停止の基準時間は、ユーザが選択する一定時間で設定されており、ユーザに負荷がかかると共に、適切な設定が困難であり、CPU（Central Processing Unit、中央処理装置）や表示装置など、未使用状態がほとんどない装置では、省電効果を全く得ることができない点と、実行するプログラムによっては、そのスループットを大きく損ねてしまう点である。本発明の目的は、これら従来技術の課題を解決し、CPUなどのように、常時使用されている装置に対しても有効な省電化を行ない、コンピュータシステムの省電制御を効率良く行なうことを可能とするコンピュータ利用機器の省電制御システムを提供することである。

【0007】

【課題を解決するための手段】上記目的を達成するため、本発明のコンピュータ利用機器の省電制御システムは、（１）コンピュータを利用した機器の省電制御システムであり、システムを構成するそれぞれの回路を、通常の動作モードから、より消費電力の低い省電動作モー

ドに切替え、システムの動作電力の低減をはかるコンピュータ利用機器の省電制御システムにおいて、プログラムから、所定の標準入力装置における入力データの発生の有無の問い合わせが要求された回数であるセンス回数を計数して記憶するセンス回数記憶部と、プログラムからの標準入力装置の制御以外の要求、もしくは、定常的に繰返し発生する割込み以外の標準入力装置を含む周辺装置からの割込み発生時に、センス回数記憶部で記憶しているセンス回数をゼロ回にリセットするセンス回数リセット部と、所定の時間間隔で繰り返される割込み時に、センス回数記憶部で記憶しているセンス回数を読み取り、このセンス回数の所定の単位時間あたりの回数であるセンス時間密度を測定して記憶するセンス時間密度記憶部と、このセンス時間密度記憶部で記憶しているセンス時間密度の、それぞれの値の所定の単位時間内における発生頻度を測定し、この発生頻度の測定結果で、所定の発生頻度値を超えているセンス時間密度を、プログラムが標準入力装置の入力データ発生を待ち続けているアイドル状態を示すアイドル状態センス時間密度として抽出して記憶するアイドル状態特定部と、センス時間密度記憶部で記憶しているセンス時間密度と、アイドル状態特定部で記憶しているアイドル状態センス時間密度との一致を照合して、アイドル状態の検出を行なうアイドル状態検出部と、このアイドル状態検出部によるアイドル状態の検出に基づき、通常の動作モードから省電動作モードへの切り替えを行なう省電動作モード設定制御部と、省電動作モード中における、プログラムからの標準入力装置の制御以外の要求、もしくは、定常的に繰返し発生する割込み以外の標準入力装置を含む周辺装置からの割込み発生時に、省電動作モードを通常の動作モードへ復帰させる通常動作モード設定制御部とを設けることを特徴とする。また、（２）上記（１）に記載のコンピュータ利用機器の省電制御システムにおいて、プログラムから、所定の標準入力装置における入力データの発生が要求された時にこの入力データが存在しない場合、この入力データが発生するまでを、省電動作モード設定部を用いて、省電動作モードに切り替える標準入力制御部を設けることを特徴とする。また、（３）上記（１）、もしくは、（２）のいずれかに記載のコンピュータ利用機器の省電制御システムにおいて、現在実行中のプログラムからのプログラム実行終了要求に基づき、アイドル状態特定部で記憶しているアイドル状態センス時間密度を、終了を要求したプログラムに付与して終了させるプログラム終了制御部を設けることを特徴とする。また、（４）上記（１）から（３）のいずれかに記載のコンピュータ利用機器の省電制御システムにおいて、省電動作モード設定制御部を用いて、省電動作モードに切り替える時、システムを構成するそれぞれの回路を、所定の順序、および、所定の時間間隔で、省電動作モードに切り替える省電動作モードフェーズ制御部と、通常動作モー

ド設定制御部を用いて、省電動作モード中から通常の動作モードへ復帰する時、システムを構成するそれぞれの回路を、所定の順序、および、所定の時間間隔で、通常の動作モードに復帰させる通常動作モードフェーズ制御部とを設けることを特徴とする。また、(5)上記

(1)から(4)のいずれかに記載のコンピュータ利用機器の省電制御システムにおいて、実行対象の任意のプログラムに対応して、少なくとも、センス時間密度記憶部で用いるセンス時間密度の測定用の所定の単位時間と、アイドル状態特定部で用いる、センス時間密度の発生頻度の測定用の所定の単位時間、および、アイドル状態センス時間密度の抽出用の所定の発生頻度値と、省電動作モードフェーズ制御部で用いる、所定の順序、および、所定の時間間隔と、通常動作モードフェーズ制御部で用いる、所定の順序、および、所定の時間間隔とを含む省電制御条件の設定値を変更する省電制御条件設定制御部を設けることを特徴とする。

【0008】

【作用】本発明においては、実行対象のプログラムから、キーボードなど、所定の標準入力装置における入力データの発生との問い合わせ(センス)が要求され、子のセンスを繰り返しているだけで、特に有機的な動作を行っていないアイドル状態を検出する。そして、そのアイドル状態の間、CPUなどを、省電動作モードに設定する。このように、実行対象のプログラムに対応して、システムの消費電力を低減する。また、実行対象のプログラムから、標準入力装置からの入力を要求された場合には、標準入力装置からの入力データがなければ、直ちに、省電動作モードに設定する。このことにより、システムの消費電力の低減をさらに効率良く行なうことができる。また、プログラムの終了時に、このプログラムに対応するアイドル状態のセンス時間密度を、終了を要求したプログラムに付与して終了する。そして、再実行する時に、このアイドル状態のセンス時間密度を利用し、再実行の直後から省電動作モードに設定することができる。また、消費電力をさらに低減することができる。また、アイドル状態の継続時に、各装置別に、省電動作モードから通常動作モードへの復帰時間が短い順序で、段階的に、省電動作モードに設定する。このことにより、アイドル状態が長く続くほど、システムの消費電力を低減でき、かつ、スループットの損失を効率良く抑えることができる。また、プログラム個別に、省電化の効果が最大になるように最適化された各種の条件の設定ができ、さらに、システムの消費電力を効率良く低減することができる。

【0009】

【実施例】以下、本発明の実施例を、図面により詳細に説明する。図5は、本発明のコンピュータ利用機器の省電制御システムの本発明に係わるアイドル状態の処理動作の一実施例を示すフローチャートである。図5

(a)、(b)における本実施例は、プログラムの、標準入力装置からの入力(以下、コンソール入力と記載)に係わる処理動作を示すものであり、図中で、太枠内の処理は、BIOS(Basic Input/Output System、オペレーティングシステム中のハードウェアに依存する制御プログラム)で行なう処理であり、その他は、実行対象のプログラムに基づき行なわれる処理である。

【0010】図5(a)は、コンソール入力に先だつて、コンソール入力が発生したか否かを問い合わせる(コンソール入力センス処理)ものであり、ほとんどのプログラムで採用されるものである。一般に、BIOSのコンソール入力センス処理では、入力データが発生するまで待ち続けてしまうので、入力が発生しなければ、別の処理を行ないたい時に採用される技術である。尚、コンソールの入力がない時に行なう例として、通信ポートや、ポインティングデバイス(マウスなど)のセンス処理などが代表的である。

【0011】図5(a)において、BIOSによるコンソール入力のセンス処理で(ステップ1501)、コンソール入力があれば(ステップ1502)、BIOSは、コンソール入力処理を行ない(ステップ1504)、プログラムは、この入力に対応する処理を行なう(ステップ1505)。しかし、ステップ1502において、コンソール入力がない場合は、プログラムは、入力無時の処理を行なう(ステップ1503)。この図5(a)では、プログラムがコンソール入力の発生を待ち続けている状態、すなわち、アイドル状態が、L1のループとして現われる。もし、コンソール入力が発生しなければ、コンソール入力のセンス処理のみが行なわれ続けるからである。このセンス処理は、入力が発生した時に、通常動作モードに戻れば良いので、L1のループを停止して、省電動作モードに入ることができる。

【0012】一方、図5(b)は、標準入力装置だけを入力装置とみなす場合に採用されるものである。ここでは、アイドル状態が、BIOSによるコンソール入力処理そのもののの中に、ループL2として現われる。このループL2内で行なわれていることは、図5(a)におけるループL1と本質的には変わらないものであり、ループL1の場合と同様に、省電動作モードに入ることができる。

【0013】図5(a)と図5(b)の処理の大きな違いは、アイドル状態を示すループが、前者は、プログラム内にあり、後者は、BIOS内にあることである。BIOSから見れば、図5(b)におけるループL2は、自身の管理下にあるので、アイドル状態の検知は極めて容易である。しかし、図5(a)のループL1は、BIOSの管理下ではないため、検出が非常に困難である。以下、このようなループL1の検出を効率良く行なうことを可能とする本発明に係わるコンピュータ利用機器の

省電制御システムに関して、次の図1を用いて説明する。

【0014】図1は、本発明を施したコンピュータ利用機器の省電制御システムの本発明に係わる構成の第1の実施例を示すブロック図である。本図において、6は、所定の標準入力装置としてのコンソール入力装置、26は、オペレーティングシステムとアプリケーションプログラムからなるプログラム（図中、Prog.と記載）、24は、本発明のセンス回数記憶部およびセンス時間密度記憶部そしてアイドル状態検出部としての処理制御を行なうコンソール入力センス制御部であり、プログラム26から要求されるコンソール入力センス回数を計数して、コンソール入力センス回数記憶部19で記憶し、かつ、コンソール入力センス回数の単位時間あたりの密度（コンソール入力センス時間密度）を測定し、このコンソール入力センス時間密度に基づき、アイドル状態を検出する。また、本図において、7は、本発明のセンス回数リセット部としての処理制御を行なう割込み制御部であり、タイマ発振装置5からの定常的に発生するタイマ割込み以外の割込みに基づき、コンソール入力センス回数記憶部19で記憶している問い合わせ回数をゼロにリセットする。

【0015】また、本図において、10は、本発明のアイドル状態特定部およびセンス時間密度記憶部としての処理制御を行なうタイマ割込み制御部であり、タイマ発振装置5による定常的なタイマ割込み時に、コンソール入力センス回数記憶部19で記憶しているコンソール入力センス回数の所定の単位時間当たりの値であるセンス時間密度を測定し、それぞれのセンス時間密度の発生頻度を測定し、所定の発生頻度値を超えているセンス時間密度を、アイドル状態を示すアイドル状態センス時間密度として抽出して、アイドル状態のコンソール入力センス時間密度記憶部14で記憶する。

【0016】また、本図において、4は、タイマ割込み制御部10で測定するそれぞれのセンス時間密度の発生頻度を記憶するコンソール入力センス時間密度の発生頻度記憶部で、15は、タイマ割込み制御部10、および、コンソール入力センス制御部24によるコンソール入力センス時間密度の測定に用いる単位時間や、発生頻度の測定に用いる所定の時間などを設定する時間計測用タイマカウンタ記憶部である。例えば、コンソール入力センス制御部24は、本発明のアイドル状態検出部として、コンソール入力センス回数記憶部19で記憶しているコンソール入力センス回数を、時間計測用タイマカウンタ記憶部4に設定した所定の単位時間で読み取り、コンソール入力センス時間密度（センス時間密度）を測定し、この測定したコンソール入力センス時間密度と、アイドル状態のコンソール入力センス時間密度記憶部14で記憶したコンソール入力センス時間密度との一致を照合して、アイドル状態の検出を行なう。

【0017】本図において、21は、コンソール入力センス制御部24によるアイドル状態の検出に基づき、後述の図4における省電動作モードフェーズ制御情報記憶部を用いて、それぞれの装置を、所定の順序、および、所定の時間間隔で、通常の動作モードから省電動作モードへの切り替えを行なう省電動作モード設定制御部、12は、本発明の通常動作モードフェーズ制御手段としての制御動作も行ない、タイマ割込み制御部10の動作に用いるタイマ割込み以外の割込みの発生に伴い、省電動作モード設定制御部21で設定したそれぞれの装置の省電動作モードを、所定の順序で、所定の時間間隔で、通常の動作モードへ復帰させる通常動作モード設定制御部、17は、現在の動作モードを記憶する現動作モード記憶部である。

【0018】尚、本図において、Aは、後述の図4で示す条件記憶部28に、B、C、Dは、それぞれ、後述の図2で示すコンソール入力割込み制御部11、コンソール入力データ記憶部16、コンソール入力制御部25に、E、F、Gは、それぞれ、後述の図3で示す表示・印刷制御部8および外部記憶制御部9、プログラムロード/セーブ制御部13、現実行プロセス記憶部18に、そして、Hは、同じく後述の図3で示す印刷装置2および外部記憶装置3に接続されている。このように、図1～図4により本発明に係わる省電制御システムの全体の構成が示されるものとして、順次図1～図4を用いて説明する。

【0019】図1に示す構成により、本第1の実施例の省電制御システムでは、実行対象のプログラムからのコンソール入力センス要求に対応して、コンソール入力センスのみを行なっているだけのアイドル状態を、このプログラムに固有に検出して、そのアイドル状態の間、CPUも含めて、省電動作モードに設定し、消費電力を効率良く低減する。

【0020】以下、本第1の実施例の省電制御システムの処理動作を説明するが、本実施例では、コンピュータ利用機器は、以下の特徴を持つものとする。

(ア) CPUは、スタンバイ、クロックダウン、ホルトモードなどの従来技術により、プログラマブルに省電動作モードとなる。

(イ) CPUが省電動作モード中でも、内部レジスタ、および、RAM(Random Access Memory、ラム)の内容は保持しておくことができる。

(ウ) CPUが省電動作モード中にある時に装置からの割込みが発生すれば、自動的に省電動作モードを即刻解除して通常の動作モードに移行することができる。

(エ) CPU以外の装置では、省電動作モード、および、通常動作モードの設定は、プログラマブルに行なうことができる。

【0021】図5(a)のルーブル1を実行中であるとすれば、コンソール入力がないまま、プログラム26

10

20

30

40

50

が、コンソール入力センサ制御部24に対し、センサ要求をし続けていることになる。コンソール入力センサ制御部24では、プログラム26からコンソール入力センサ要求が発生する度に、その回数をコンソール入力センサ回数記憶部19に記憶する。また、所定の時間を測定するために、一定周期の発信が、タイマ発信装置5で発生し、これが、割込み制御部7を経由して、CPUに割込みを発生させ、タイマ割込み制御部10が起動される。この時、同時に、割込み制御部7から通常動作モード設定制御部12に、通常動作モードとするように要求するため、タイマ割込み制御部10は、通常の動作が保証される。

【0022】タイマ割込み制御部10では、コンソール入力センサ回数記憶部19に記憶されるコンソール入力センサ回数を、時間計測用タイマカウンタ記憶部15のカウンタ値を利用して、単位時間毎に取り出し、コンソール入力センサ回数の時間密度を求め、さらに、この時間密度の値別に、発生頻度を測定して、コンソール入力センサ時間密度の発生頻度記憶部4に記憶することを、一定時間続けた後、この記憶された発生頻度を解析する。尚、この処理動作は、後述の図8、9により詳細に説明する。

【0023】本第1の実施例では、タイマ割込み制御部10によるこの発生頻度の解析を、コンソール入力センサ回数の時間密度がある一定値以上で、かつ、その発生頻度がある一定値以上、という条件を満足したものとしている。すなわち、図5(a)のループL1の実行中であることを示すものとして、ここでは、集中的に発生したコンソール入力センサ回数の時間密度を抽出する。尚、このコンソール入力センサ回数の時間密度の抽出に関しては、様々な技術が考えられるので、後述の図8、9で示す実現例は、その一例である。このようにして、タイマ割込み制御部10により、集中的に発生したコンソール入力センサ回数の時間密度を抽出したならば、この時間密度を、アイドル状態のものとみなし(複数種類でも良い)、アイドル状態のコンソール入力センサ時間密度記憶部14に記憶する。

【0024】コンソール入力センサ制御部24は、上述のコンソール入力センサ回数の記憶以外に、タイマ割込み制御部10と同様にして、コンソール入力センサ回数の時間密度を求め、この時間密度と同じ値の時間密度を、アイドル状態のコンソール入力センサ時間密度記憶部14から探し出す。もしあれば、現在、プログラム26は、図5(a)のループL1を実行中と判断できるので、コンソール入力センサ制御部24は、省電動作モード設定制御部21に、省電動作モードの設定を要求する。そして、省電動作モード設定制御部21は、動作モードを、省電動作モードに切替る。また、アイドル状態のコンソール入力センサ時間密度記憶部14に、該当するコンソール入力センサ回数の時間密度がなければ、ル

ープL1であるとは判断できないので、再び、コンソール入力センサ回数の時間密度を求め、同様の作業を繰り返す。

【0025】このように、本第1の実施例では、プログラム26のアイドル状態を、動的に検出する。このことにより、コンソール入力センサ処理を行なうそれぞれのプログラムが、本実施例の省電制御システムの動作を関知していない場合でも、効率良く省電動作モードの設定を行なうことができる。

【0026】以上のように、プログラム26のアイドル状態を検出して、省電動作モードに設定し、システムの省電化を行なうが、この中で、コンソール入力センサ回数は、単純に、プログラム26からコンソール入力センサ制御部24が使用された回数ではない。タイマ割込み制御部10の割込み動作を制御するタイマ発振装置5を除く周辺装置、すなわち、コンソール入力装置6や、後述の図3に示す印刷装置2と外部記憶装置3などからの割込みが発生した時、および、同じく後述の図3に示す周辺入出力サービス制御部22や実行プロセス制御部23のように、コンソール入力関係以外の制御ブロックが、プログラム26から呼ばれた時には、コンソール入力センサ回数記憶部19に記憶しているコンソール入力センサ回数を、「0」(ゼロ)にリセットして、コンソール入力センサ回数を最初から記憶させる。これは、図5(a)におけるループL1の入力無時の処理内で、表示などを行なっているプログラムに対応するためである。すなわち、もし、このリセットを行わなければ、表示が止まってしまうことが予想される。尚、このようなリセット動作に関しては、後述の図12、13で詳細に説明する。

【0027】次に、図5(b)の動作を行なうプログラムに対応する省電制御システムについての説明を、図2を用いて行なう。図2は、本発明を施したコンピュータ利用機器の省電制御システムの本発明に係わる構成の第2の実施例を示すブロック図である。本第2の実施例は、図1における第1の実施例の省電制御システムに基づくものであり、本図において、6、24、26のそれぞれは、図1で示したコンソール入力装置、コンソール入力センサ制御部、プログラムであり、11は、コンソール入力装置6からのコンソール入力割込みを制御するコンソール入力割込み制御部、16は、コンソール入力割込み制御部11を介して入力されたコンソール入力装置6からのコンソール入力を記憶するコンソール入力データ記憶部、20は、コンソール入力データ記憶部16に記憶したコンソール入力のデータ変換を行なうコンソール入力データ変換部、そして、25は、図1の省電動作モード設定制御部21を介して、プログラム26からの要求に応じて、コンソール入力装置6からの入力データが発生するまでを、アイドル状態に切り替える本発明の標準入力制御部としての処理制御を行なうコンソール

入力制御部である。

【0028】本第2の実施例においては、図5(b)に示すように、プログラム26が、コンソール入力制御部25に、コンソール入力が発生するまで、制御を完全に譲っており、コンソール入力制御部25では、コンソール入力が発生していなければ、即刻、省電動作モードにして、その発生を待つだけである。このことにより、図5(b)における動作では、最大高率で省電化を行なうことができる。尚、このような第2の実施例の省電制御システムの動作に関して、後述の図18で説明する。

【0029】以上、図1、および、図2により、図5(a)、(b)のそれぞれのケースにおける省電化の動作を説明したが、次に、図3を用いて、本発明の、次の特徴である、プログラム固有の省電化を可能にする実施例に関して説明する。すなわち、アイドル状態のコンソール入力センサ時間密度は、プログラム毎に固有の値を持つことが考えられるが、プログラム内の処理部(コード部)は、何度実行されても、通常は変わることがない。従って、一度検出したアイドル状態のコンソール入力センサ時間密度を、そのプログラム固有の領域に保存しておき、再実行時に、これを利用すれば、実行直後から、省電動作モードにすることができ、さらに、消費電力を低減できる。

【0030】図3は、本発明を施したコンピュータ利用機器の省電制御システムの本発明に係わる構成の第3の実施例を示すブロック図である。本第3の実施例は、図1、および、図2における省電制御システムに基づくものであり、本図において、14、26は、それぞれ、図1で示したアイドル状態のコンソール入力センサ時間密度記憶部とプログラム、1は、CRT(Cathode Ray Tube、陰極線管)などからなる表示装置、2は、ドットプリンタなどの印刷装置、3は、プログラム26などを格納する外部記憶装置、8は、表示装置1と印刷装置2の出力動作制御を行なう表示・印刷制御部、9は、外部記憶装置3の動作制御を行なう外部記憶制御部、22は、プログラム26からの要求に対応して、表示装置1や、印刷装置2、外部記憶装置3などを、それぞれ、表示・印刷制御部8と外部記憶制御部9を介して動作制御する周辺入出力サービス制御部、23は、現在実行中のプログラムからのプログラム実行終了要求に応じて、アイドル状態のコンソール入力センサ時間密度記憶部14で記憶しているアイドル状態センサ時間密度を、終了要求したプログラムに付与して終了させる本発明のプログラム終了制御部としての実行プロセス制御部、13は、実行プロセス制御部23を介して、プログラムからの指示に基づき、プログラムのロードやセーブ動作制御を行なうプログラムロード/セーブ制御部、18は、現在実行されているプログラム番号を記憶する現実行プロセス記憶部である。

【0031】以下、プログラムの実行をプロセスの開

始、プログラムの終了をプロセスの終了と呼ぶことにし、本第3の実施例の動作の説明を行なう。また、本第3の実施例では、既に開始されたあるプロセス内から、さらに、別のプロセスを開始させる動作について説明する。まず、プロセスの開始について、既にプロセスが開始されている状態から説明する。

【0032】プログラム26が、実行プロセス制御部23に、新規プロセスの開始を要求すると、現在のプロセスで検出されたアイドル状態のコンソール入力センサ時間密度記憶部14のアイドル状態のセンサ回数の時間密度を温存した上で、プログラムロードセーブ制御部13に、該当プログラムを読み込ませる。この時、次のプロセスの終了で説明するように、その該当プログラム固有の記憶域に、アイドル状態のセンサ回数の時間密度が記憶されているので、これも、同時に読み込んで、アイドル状態のコンソール入力センサ時間密度記憶部14を書き換える。そして、現プロセスの呼び出し元に、プログラム終了アドレスをセットして、新規プロセスのプログラムエントリにジャンプする。尚、このような動作に関しては、後述の図19により詳細に説明する。

【0033】次に、このプロセスの終了について説明する。プログラム26が、実行プロセス制御部23に、現プロセスの終了を要求すると、現プロセスで検出され、アイドル状態のコンソール入力センサ時間密度記憶部14に記憶されているアイドル状態のコンソール入力センサ回数の時間密度を、現プロセスの元となるプログラム固有の記憶域に保存しておく。尚、本実施例では、後述の図22に示すように、該当プログラムファイル自身223に埋め込んでいる。次に、現プロセスの呼び出し元のアイドル状態のコンソール入力センサ回数の時間密度が、上述の説明のように、アイドル状態のコンソール入力センサ時間密度記憶部14に温存されているので、それを復帰し、呼び出しもとにジャンプする。尚、このプロセスの終了動作に関しては、後述の図20により詳細に説明する。

【0034】このように、本第3の実施例では、プロセスの開始と終了を行なっても、タイマ割込み制御部10で行なわれるアイドル状態の検出を毎回やり直すことなく、プロセス開始時から、以前にそのプロセスが終了した時の省電制御を、再び開始できるので、無駄がない。すなわち、既に一度でも、アイドル状態が検出されたプログラムであれば、二度目からは、すぐに、省電効果が期待できる。

【0035】次に、省電制御条件をプログラム毎に変更する動作、および、装置別の段階的な省電動作モードの設定に係わる省電制御システムに関して、図4を用いて説明する。図4は、本発明を施したコンピュータ利用機器の省電制御システムの本発明に係わる構成の第4の実施例を示すブロック図である。本実施例は、図1～図3の省電制御システムに基づくものであり、本図におい

10

20

30

40

50

て、10、24、26は、それぞれ、図1で示したタイマ割込み制御部、コンソール入力センサ制御部、プログラムであり、21、25は、図2で示した省電動作モード設定制御部とコンソール入力制御部、また、23は、図3で示した実行プロセス制御部、そして、27は、本発明に係わり、実行対象の任意のプログラムに対応して、コンソール入力センサ時間密度の測定用の単位時間や発生頻度の測定用の所定の時間などの省電制御条件の設定値を変更する省電制御条件の設定制御部、28は、省電制御条件の設定制御部28で設定される省電制御条件を記憶する条件記憶部、29は、省電動作モード設定制御部21が、本発明の省電動作モードフェーズ制御部としての処理動作を併せて行なうために用いる省電動作モードフェーズ制御記憶部であり、周辺装置に対する省電動作モード設定制御部21による電力供給の停止などを、予め定められた順序および時間間隔で制御するための情報が記憶されている。

【0036】図1～図3で示した省電制御システムでは、コンソール入力センサ時間密度の測定用の単位時間や、発生頻度の測定用の所定の時間、および、アイドル状態の抽出用の発生頻度など、省電制御のための幾つかの条件を用いている。例えば、それぞれの周辺装置の省電動作モードから通常動作モードへの復帰時間を、省電動作モードフェーズ制御情報記憶部29に記憶しておき、省電動作モード設定制御部21は、この条件を参照することにより、復帰時間の短い装置から、段階的に省電動作モードに設定する。このことにより、アイドル状態が長く続くほど、消費電力を低減でき、復帰に要する時間も短縮することができる。尚、このような段階的な省電動作モードの設定に関しては、後述の図17で詳細に説明する。

【0037】また、省電制御の条件を変更すれば、アイドル状態の具合や、省電化する装置の種類などを変更することができる。すなわち、プログラム26が、省電制御条件の設定制御部27へ、これらの条件の変更を要求し、省電制御条件の設定制御部27は、条件記憶部28の変更を行なう。この条件記憶部28は、省電制御に係わるタイマ割込み制御部10や、実行プロセス制御部23、コンソール入力センサ制御部24、コンソール入力制御部25などから参照され、省電制御条件が一括して変更される。このことにより、各プログラム個別に、省電化の効果が最大になるように最適化された各種の条件のもとに、省電制御を行なうことができる。尚、このような省電制御の条件の変更動作の詳細に関しては、後述の図23で説明する。

【0038】次に、図6～図23を用いて、図1～図4で示した省電制御システムのそれぞれの動作を、詳細に説明する。尚、図6～図23での記述はC言語に従うものとする。図6は、図10～図23の説明で用いられる変数の内容を示す説明図である。変数名「run_mo

de」は、現在の動作モードで、「0」で通常動作モード、「0」以外で省電動作モードとなる。初期値は「0」で、図1の現動作モード記憶部17に設定される。「process」は、現在実行中のプロセス番号で、「0」～「PROCESS_MAX-1」の値である。初期値「0」で、図3の現実行プロセス記憶部18に設定される。

【0039】「keyb_cnt」は、現在までに入力されている有効データ数で、「0」～「記憶バッファの大きさ」までの値である。初期値「0」で、図2のコンソール入力データ記憶部16に設定される。「dens_cnt」は、図1のタイマ割込み制御部10用のコンソール入力センサ回数である。初期値「0」で、図1のコンソール入力センサ回数記憶部19に設定される。

「dens_tmr」は、図1のタイマ割込み制御部10用のコンソール入力センサ時間密度の測定用タイマカウンタである。初期値「SENS_TMR_MAX」で、図1の時間計測用タイマカウンタ記憶部15に設定される。

【0040】「freq_tmr」は、コンソール入力センサ時間密度の頻度測定用タイマカウンタである。初期値「FREQ_TMR_MAX」で、図1の時間計測用タイマカウンタ記憶部15に設定される。「freq_tbl」は、コンソール入力センサ時間密度の頻度テーブルである。初期値は全て「0」で、図1のコンソール入力センサ時間密度の発生頻度記憶部4に設定される。「dens_tbl」は、アイドル状態のコンソール入力センサ時間密度テーブルである。初期値は全て

「0」で、図1のアイドル状態のコンソール入力センサ時間密度記憶部14に設定される。「sens_cnt」は、図1のコンソール入力センサ制御部24用のコンソール入力センサ回数である。初期値「0」で、図1のコンソール入力センサ回数記憶部19に設定される。

「sens_tmr」は、図1のコンソール入力センサ制御部24用のコンソール入力センサ時間密度の測定用タイマカウンタである。初期値「SENS_TMR_MAX」で、図1の時間密度計測用タイマカウンタ記憶部15に設定される。

【0041】「sleep_tmr」は、省電動作モードの各フェーズ（段階的な設定順序）を設定するためのタイマカウンタであり、初期値は「run_mode」による。図1の時間密度計測用タイマカウンタ記憶部15に設定される。「sleep_tbl」は、省電動作モードの各フェーズを設定するための制御テーブルであり、初期値は、例えば、後述の図9に示すように設定され、図4の省電動作モードフェーズ制御情報記憶部29に設定される。「active_tmr」は、省電動作モードから通常動作モードへ復帰する時の待ち時間であり、初期値は「run_mode」による。図1の時間密度計測用タイマカウンタ記憶部15に設定される。

【0042】図7は、後述の図10～図23の説明で用いられる条件定数の内容を示す説明図である。条件定数名「SENS_TMR_MAX」は、変数「dens_cnt」、「sens_cnt」を、コンソール入力センサ時間密度とみなすまでに発生するタイマ割込みの数であり、例えば、割込み間隔が10msecの時には、初期値として「2」が設定される。「FREQ_TMR_MAX」は、コンソール入力センサ時間密度を、その発生頻度計測のためにサンプリングする回数であり、初期値に、例えば、10秒間サンプリングとして、「500」が設定される。「FREQ_TBL_MAX」は、コンソール入力センサ時間密度の発生頻度計測のための時間密度値の最大値+「1」の数であり、初期値として、例えば、「100」が設定される。

【0043】「FREQ_MIN」は、アイドル状態とみなせるコンソール入力センサ時間密度の発生頻度の最小値であり、初期値は、例えば、「30」が設定される。「DENS_MIN」は、アイドル状態とみなせるコンソール入力センサ時間密度の最小値であり、初期値は、例えば、「10」が設定される。「DENS_TBL_MAX」は、アイドル状態として登録できるコンソール入力センサ時間密度の個数であり、初期値に、例えば、「8」が設定される。「PROCESS_MAX」は、温存できるプロセスの最大数であり、初期値に、例えば、「8」が設定される。「SLEEP_MAX」は、省電動作できる装置の最大数であり、初期値としては、例えば、図9(b)に示す場合には、「2」が設定される。

【0044】図8および図9は、図6における各変数のデータ構造を示す説明図である。図8において、変数「run_mode」、「process」、「keyb_cnt」は、1バイトで、また、「dens_cnt」、「dens_tmr」、「freq_tmr」は、2バイトで構成されている。「freq_tbl」は、それぞれ2バイトの要素で構成され、「(FREQ_TBL_MAX+1)×2」のサイズである。「dens_tbl」は、それぞれ2バイトの要素で構成され、「(DENS_TBL_MAX+1)×2」の項目サイズと、「PROCESS_MAX」の行サイズである。そして、「sens_cnt」と「sens_tmr」は、2バイトで構成されている。

【0045】図9(a)において、変数「sleep_tmr」、および、「active_tmr」は、2バイトで構成されている。「sleep_tbl」は、それぞれ2バイトの要素で構成され、2バイト×3の項目サイズと、「SLEEP_MAX」の行サイズである。この「sleep_tbl」において、最初の項目は、省電動作させる装置の種別コードが設定され、例えば、図9(b)に例示される、「SLEEP_MAX」が2の「sleep_tbl」では、バックライトの種別コ

ード「1」と、ビデオコントローラの種別コード「2」の2つが設定されている。

【0046】また、「sleep_tbl」の真中の項目は、装置の次に省電動作モード設定するまでの時間であり、図9(b)に例示するように、バックライトとビデオコントローラに対して、「12000」と「30000」が、すなわち、タイマ割込み間隔が10msecの時に、それぞれ、2分と5分の待ち時間が設定されている。そして、「sleep_tbl」の最後の項目は、装置の通常動作モードへ復帰する時に必要となる待ち時間であり、図9(b)に例示されるように、バックライトには「0」が設定され、復帰時間は0秒とされ、また、ビデオコントローラには「100」で、設定されたタイマ割込み間隔が10msecの時に、1秒の復帰時間が設定されている。

【0047】以下、図10～図23により、図6～図9で示した変数を用いて、図1～図4における省電制御システムの動作を説明する。

【0048】図10は、図1におけるタイマー割込み制御部の本発明に係わる処理動作の一実施例を示すフローチャートである。本図は、コンソール入力センサ時間密度の測定と頻度テーブルの更新の手順を示すものであり、まず、図1の現動作モード記憶部17の記憶内容に基づき、現在のモードが、通常動作モード中か省電動作モード中であることを確認する(図中、「run_mode」?) (ステップ101)。省電動作モード中であれば(!=0)、後述の図15で示すように、次の装置の省電動作モードへの切替処理のための、図1の時間計測用タイマカウンタ記憶部15に記憶されている省電動作モードの設定制御用のタイマカウンタの更新を行なう。すなわち、省電動作モードの各フェーズを設定するためのタイマカウンタを確認し、(図中、sleep_tmr?) (ステップ102)、タイムアウトしていなければ(!=0)、ディクリメントし(図中、--sleep_tmr) (ステップ103)、タイムアウトしていれば(==0)、省電動作モードから通常動作モードへの復帰時の待ち時間を確認する(図中、active_tmr?) (ステップ104)。この時間がタイムアウトしていなければ(!=0)、ディクリメントし(図中、--active_tmr) (ステップ105)、処理を終了する。

【0049】ステップ101において、通常動作モード中であれば(==0)、まず、図1のコンソール入力センサ制御部24による後述の図14で示すコンソール入力センサ制御での参照に用いるコンソール入力センサ時間密度測定用の第1のタイマカウンタ更新を行なう。すなわち、図1の時間計測用タイマカウンタ記憶部15に記憶してある第1のタイマカウンタを確認し(図中、sens_tmr?) (ステップ106)、タイムアウトしていなければ(!=0)、ディクリメントする(図

中、`---sens_tmr`) (ステップ107)。

【0050】このタイマカウンタ更新処理の後に、本タイマ割込み制御部で用いるコンソール入力センサ時間密度測定用の第2のタイマカウンタを、図1の時間計測用タイマカウンタ記憶部15を参照して確認する(図中、`dens_tmr?`) (ステップ108)。タイムアウトしていなければ(`!=0`)、何もせず、次の割込みを待つ。タイムアウトしていれば(`=0`)、まず、次の制御に備えて、この第2のタイマカウンタに初期値を設定する(図中、`dens_tmr=SENS_TMR_MAX`) (ステップ109)。そして、この時のコンソール入力センサ回数を、図1のコンソール入力センサ回数記憶部19を参照して確認する(図中、`dens_cnt?`) (ステップ110)。尚、このコンソール入力センサ回数(`dens_cnt`)は、後述の図14で示すように、図1のコンソール入力センサ制御部24により登録される。

【0051】このコンソール入力センサ回数に対応する図1のコンソール入力センサ時間密度の発生頻度記憶部4内の頻度テーブルの要素を、あるいは、このコンソール入力センサ回数が頻度発生のための時間密度値の最大値+1より大きければ(`>=FREQ_TBL_MAX`)、頻度発生のための時間密度値の最大値のコンソール入力センサ回数に対応する頻度テーブル(図中、`dens_cnt=FREQ_TBL_MAX-1`) (ステップ111)の要素をインクリメントする(図中、`++freq_tbl[dens_cnt]`と記載) (ステップ112)。そして、次の頻度テーブルの更新のために、このコンソール入力センサ回数を「0」にリセットする(図中、`dens_cnt=0`と記載) (ステップ113)。

【0052】このような動作を、所定のサンプリング回数分繰り返す。すなわち、コンソール入力センサ時間密度の頻度測定用の第3のタイマカウンタを、図1の時間計測用タイマカウンタ記憶部15を参照して確認し(図中、`freq_tmr?`) (ステップ114)、この第3のタイマカウンタがタイムアウトするまで(`=0`)繰り返す。この第3のタイマカウンタがタイムアウトすれば(`=0`)、次のサンプリングのために、この第3のタイマカウンタに初期値を設定し(図中、`freq_tmr=FREQ_TMR_MAX`) (ステップ115)、その後、ステップ112で更新した頻度テーブル(`freq_tbl`)を解析して、次の図11で詳細を説明するように、アイドル状態のコンソール入力センサ時間密度の抽出と、図1のアイドル状態のコンソール入力センサ時間密度記憶部14への登録を行なう(ステップ116)。

【0053】図11は、図1におけるタイマ割込み制御部の頻度テーブルの解析とアイドル状態のコンソール入力センサ時間密度の抽出および登録処理動作の一実施

例を示すフローチャートである。まず、現在実行中のプロセス(プログラム)のアイドル状態のコンソール入力センサ時間密度テーブルのアドレスの設定を、後述の図15で示すようにして行なう(図中、`addr=現プロセスのdens_tbl`アドレス) (ステップ201)。次に、最初のコンソール入力センサ時間密度を取り出し(図中、`i=0`) (ステップ202)、その値と、所定の値、すなわち、アイドル状態とみなせるコンソール入力センサ時間密度の最小値(`DENS_MIN`、本実施例では「10」)との比較を行なう(図中、`i?`) (ステップ203)。

【0054】ここでは、`i=0`で、所定の値「10」、より小さいので(`i<DENS_MIN`)、無条件にアイドル状態とは見なさず、図1のコンソール入力センサ時間密度の発生頻度記憶部4内の頻度テーブルを初期化して(図中、`freq_tbl[i]=0`) (ステップ204)、次のコンソール入力センサ時間密度に対する処理に移る(図中、`++i`) (ステップ205)。次も`i=1`であり、同様な処理となる。このような処理を繰返し、`i=10`になれば、ステップ203において、所定の値と等しくなり(`i>=DENS_MIN`)、コンソール入力センサ時間密度「`i=10`」が、アイドル状態を示す可能性ありとして、図1のコンソール入力センサ時間密度の発生頻度記憶部4内の頻度テーブルの対応する要素を参照する(図中、`freq_tbl[i]`) (ステップ206)。

【0055】例えば、コンソール入力センサ時間密度「10」の発生頻度が、所定の頻度値、すなわち、アイドル状態とみなせるコンソール入力センサ時間密度の発生頻度の最小値(`FREQ_MIN`)より小さければ、次のコンソール入力センサ時間密度「11」の処理に移り、大きければ、図1のアイドル状態のコンソール入力センサ時間密度記憶部14に、このアイドル状態のコンソール入力センサ時間密度「10」を登録できるか否かを調査する(図中、`addr[0]?`)。

【0056】すなわち、現プロセスのアイドル状態のコンソール入力センサ時間密度テーブル(`dens_tbl`)の要素「0」で示されている、アイドル状態として既に登録してあるコンソール入力センサ時間密度の個数が、所定の、アイドル状態として登録できるコンソール入力センサ時間密度の個数(`DENS_TBL_MAX`)より小さく、図1のアイドル状態のコンソール入力センサ時間密度記憶部14の登録テーブルに空きがあることを確認し(ステップ207)、その後、コンソール入力センサ時間密度「`i=10`」が、既に登録テーブル内に存在するか否かを探索し(ステップ208)、未登録であることを確認する(ステップ209)。そして、アイドル状態のコンソール入力センサ時間密度の登録と(図中、`addr[addr[0]]=i`)、登録済みのコンソール入力センサ時間密度の個数をインクリ

メントと(図中、++addr[0])を行ない(ステップ210)、次のコンソール入力センサ時間密度「11」の処理に移る(ステップ204、205)。

【0057】このようにして、図1のタイマ割込み制御部10は、例えば、10msecの割込み間隔で、図1のコンソール入力センサ時間密度の発生頻度記憶部4に記憶した内のコンソール入力センサ時間密度の内、発生頻度の高いものを、図1のアイドル状態のコンソール入力センサ時間密度記憶部14に登録する。そして、このアイドル状態のコンソール入力センサ時間密度記憶部14の登録内容に基づき、後述の図14に示すように、図1のコンソール入力センサ制御部24により、省電動作モードの設定を行なう。尚、このような図1のタイマ割込み制御部10の処理で用いるコンソール入力センサ回数(dens_cnt)は、次の図12、および、図13で示すように、図1のコンソール入力装置6、および、図3の外部記憶装置3や印刷装置2などの周辺入出力装置からの割込み発生により、「0」にリセットされる。

【0058】図12は、図1における省電制御システムのコンソール入力装置からの割込みに基づく処理動作の一実施例を示すフローチャートである。図1におけるコンソール入力装置6からの割込みがあれば、図1のコンソール入力割込み制御部11は、割込みステータスを確認して(ステップ301)、エラーがなければ、次にコンソール入力ステータスを確認する(ステップ302)。コンソール入力ステータスが有効であれば、コンソール入力データを取り出し、図1のコンソール入力データ記憶部16の記憶バッファに記憶し(ステップ303)、現在までに入力されている有効データ数(0〜記憶バッファの大きさまで)をインクリメントする(図中、++keyb_cnt)(ステップ304)。そして、図1のコンソール入力センサ回数記憶部19のコンソール入力回数のリセットと(dens_cnt=0、sens_cnt=0)、図1の現動作モード記憶部17の記憶される現在の動作モードを、通常動作モード中に設定する(run_mode=0)(ステップ305)。

【0059】図13は、図1における省電制御システムの印刷装置および外部記憶装置からの割込みに基づく処理動作の一実施例を示すフローチャートである。図1の印刷装置2や、外部記憶装置3からの割込みに基づき、図1のコンソール入力センサ回数記憶部19のコンソール入力回数のリセットと(dens_cnt=0、sens_cnt=0)、図1の現動作モード記憶部17に記憶される現在の動作モードを、通常動作モード中に設定し(run_mode=0)(ステップ401)、図1の印刷装置2や、外部記憶装置3からの割込み処理を行なう(ステップ402)。

【0060】図14は、図1におけるコンソール入力セ

ンス制御部の本発明に係わる処理動作の一実施例を示すフローチャートである。本図は、プログラムから要求されたコンソール入力センサ回数の時間密度を測定し、その時間密度に基づき、省電モードの設定を制御する手順を示すものであり、まず、図1のコンソール入力装置6からの入力があるか否かを調査する(図中、keyb_cnt?)(ステップ501)。入力があれば、図12で示すコンソール入力割込み処理に移り、入力がなければ、図1のコンソール入力センサ回数記憶部19の記憶内容、すなわち、コンソール入力センサ回数をインクリメントとする(図中、++dens_cnt)(ステップ502)。尚、このコンソール入力センサ回数は、図10におけるステップ110の処理で参照されるものであり、かつ、図12、および、図13に示した割込み処理により、「0」にリセットされる。

【0061】次に、図1の時間計測用タイマカウンタ記憶部15のコンソール入力センサ時間密度の測定用のタイマカウンタの値を確認する(図中、sens_tmr?)(ステップ503)。尚、このタイマカウンタ(sens_tmr)の値は、図10のステップ107の処理で、初期値(SENS_TMR_MAX、コンソール入力センサ回数を、コンソール入力センサ時間密度とみなすまでに発生するタイマ割込みの数で、本実施例では、割込み間隔が10msecで、「2」とする)から、順次にデクリメントされている。このタイマカウンタ(sens_tmr)が「0」になるまで、コンソール入力センサ回数(sens_cnt)のインクリメント(ステップ504)と、ステップ502のコンソール入力センサ回数(dens_cnt)のインクリメントを繰り返す。

【0062】コンソール入力センサ時間密度の測定用のタイマカウンタ(sens_tmr)が「0」になれば、以下、現在のコンソール入力センサ時間密度がアイドル状態のものか否かを判定し、省電動作モードの設定を行なう。すなわち、図11におけるステップ201の処理と同様に、現在実行中のプロセス(プログラム)のアイドル状態のコンソール入力センサ時間密度テーブルのアドレスの設定を、後述の図13で示すようにして行なう(図中、addr=現プロセスのdens_tblアドレス)(ステップ505)。そして、設定したアドレスから、これまでにカウントしたコンソール入力センサ回数(sens_cnt)と同一の値のものを探索する(ステップ506)。同一値があれば(ステップ507)、後述の図15で示す省電動作モードの設定制御を行なう(ステップ508)。同一値がなければ、また、省電動作モードの設定が停止されたならば、次の省電動作モードの設定制御のために、コンソール入力センサ回数と、コンソール入力センサ時間密度の測定用のタイマカウンタをそれぞれ初期値に戻す(図中、sens_cnt=0、sens_tmr=SENS_TMR_MA

X) (ステップ509)。

【0063】図15は、図3の省電制御システムにおける実行プロセス制御部の本発明に係わる処理動作の一実施例を示すフローチャートである。本図は、図1におけるアイドル状態のコンソール入力センサ時間密度記憶部14に記憶される内容を、実行される複数のプログラムのそれぞれに対応させるために、現在のプロセスのアイドル状態のコンソール入力センサ時間密度テーブル(dens_tbl)のアドレスを取得する手順を示すものである。すなわち、アイドル状態として登録できるコンソール入力センサ時間密度の個数(DENS_TBL_MAX、本実施例では「8」)に「1」を加えた値を2倍(各データが2バイトのため)して、現在実行中のプロセス番号(「0」～「PROCESS_MAX(温存できるプロセスの最大数)-1」)を掛けた値を、テーブルの先頭アドレスに加える(図中、addr=dens_tbl先頭アドレス+(DENS_TBL_MAX+1)×2×process)(ステップ601)。このようにして取得した現プロセスのアドレスに基づき、図1のタイマ割込み制御部10、および、コンソール入力センサ制御部24は、図1のアイドル状態のコンソール入力センサ時間密度記憶部14のテーブル(dens_tbl)を参照することができる。

【0064】図16は、図1および図3における省電制御システムのプログラムからの周辺装置の要求に基づく処理動作の一実施例を示すフローチャートである。図3の印刷装置2や外部記憶装置3などのように、図1のコンソール入力装置6以外の装置に対する制御が、図3のプログラム26から呼ばれた時には、図3の周辺入出力サービス制御部22により、図1のコンソール入力センサ回数記憶部19のコンソール入力回数のリセットと(dens_cnt=0、sens_cnt=0)、図1の現動作モード記憶部17の記憶される現在の動作モードを、通常動作モード中に設定して(run_mode=0)(ステップ701)、図3の印刷装置2や、外部記憶装置3などに対する制御処理を行なう(ステップ702)。

【0065】図17は、図1および図4における省電制御システムの本発明に係わる省電動作モードの設定制御手順の一実施例を示すフローチャートである。本図は、図14のステップ508における省電動作モードの設定制御の処理内容を示しており、まず、省電動作モードから通常動作モードへの復帰時の待ち時間を設定する(図中、active_tmr=0)(ステップ801)。尚、ここでは、CPUのみの省電動作時の通常動作モードへの復帰時間「0」を設定する。次に、現在の動作モードを示す値をインクリメントした値と、所定の値、すなわち、省電動作できる装置の最大数(SLEEP_MAX、本実施例では、図9(b)に示すように「2」)との比較を行なう(図中、++run_mode?)

(ステップ802)。

【0066】例えば、通常動作モード中からの最初の省電動作モードの設定であれば、現在の動作モードを示す値は「0」であり、これをインクリメントした値は「1」となり、所定の値「2」(SLEEP_MAX)よりも小さいので、この現在の動作モード「1」に基づき、省電動作モードの各フェーズを設定するための制御テーブル(sleep_tbl)を参照して、省電動作モードの各フェーズを設定するためのタイマカウンタ(sleep_tmr)を設定する(ステップ803)。本実施例では、制御テーブル(sleep_tbl)において、図9(a)、(b)で示すように、次に省電動作モード設定するまでの時間は、中間に登録されているので、「sleep_tbl[(run_mode-1)×3+1]」で求められる位置の値を、タイマカウンタ(sleep_tmr)に設定する。

【0067】そして、CPUを省電動作モードに設定する(ステップ804)。この状態で、省電動作モードが継続すると(ステップ805)、図10におけるステップ103の処理により、ステップ803で設定したタイマカウンタ(sleep_tmr)が、各割込み処理毎にデクリメントされる。このタイマカウンタ(sleep_tmr)が、「0」になれば(図中、sleep_tmr?) (ステップ806)、図9(a)で示す省電動作モードの各フェーズを設定するための制御テーブル(sleep_tbl)に基づき、「sleep_tbl[(run_mode-1)×3]」で指示される装置を、省電動作モードに設定する(ステップ807)。例えば、図9(b)に示す制御テーブル(sleep_tbl)に基づけば、CPUの省電動作モード設定の2分後に、バックライトの省電動作モードの設定を行なう。

【0068】この省電動作モードの設定が終了すれば、設定された装置の、省電動作モードから通常動作モードへの復帰時の待ち時間の設定を行なう(図中、active_tmr=sleep_tbl[(run_mode-1)×3+2]) (ステップ808)。ここでは、図9(b)に示されるように、バックライトの復帰時間「0」が設定される。次に、ステップ802に戻り、現在の動作モードを示す値「1」を、「2」にインクリメントし、以下同様にして、ステップ803～808の処理を行なう。例えば、図9(b)に示す制御テーブル(sleep_tbl)に基づき、バックライトの省電動作モード設定の5分後に、復帰時間1秒で、ビデオコントローラの省電動作モードの設定を行なう。

【0069】この設定後に、ステップ802では、インクリメントした動作モードの値が「3」となり、図9(b)に基づく所定の値「2」(SLEEP_MAX)を超えるので、DRAMを残して、全ての電源をオフにして、サスペンド状態に設定する(ステップ809)。

また、ステップ805において、ハードウェア割込みがあり、省電動作モードから通常動作モードへの復帰が発生した場合には、これまでに省電動作モードに設定を完了した全ての装置（図中、`sleep_tbl[0] ~ sleep_tbl[(run_mode-2)×3]`）を、通常動作モードに設定する（ステップ810）。この時、ステップ808で設定されたそれぞれの装置の復帰時間の合計時間まで待った後に（図中、`active_tmr?`）（ステップ811）、図14のステップ509に移る。尚、各装置の復帰時間（`active_tmr`）のディクリメントは、図10におけるステップ105の処理で行なわれている。

【0070】図18は、図2の省電制御システムにおけるコンソール入力制御部の本発明に係わる処理動作の一実施例を示すフローチャートである。本図は、プログラムがコンソール入力センスを要求しない動作を行なう場合において、図2のコンソール入力制御部25の処理動作を示すものである。コンソール入力の発生の有無を確認し（図中、`keyb_cnt?`）（ステップ901）、コンソール入力が無ければ（`=0`）、図17に示す省電動作モードの設定制御を行なう（ステップ902）。もし、コンソール入力があれば（`!=0`）、コンソール入力センサ回数のリセット（図中、`sens_cnt=0`）と、コンソール入力センサ時間密度測定用のタイマカウンタの初期化（図中、`sens_tmr=SENS_TMR_MAX`）を行ない（ステップ903）、入力されたデータの取り出しと、図1のコンソール入力データ変換部20を介してのデータ変換を行なう（ステップ904）。このように、コンソール入力センサを要求しないプログラムの場合には、コンソール入力

が発生していなければ、即刻、省電動作モードにして、その発生を待つだけであり、最大の効率で省電化を行なうことができる。

【0071】以上、図10～図18により、コンソール入力センサを要求するプログラムと、コンソール入力センサを要求しないプログラムの、それぞれのケースにおいて、本発明に係わる省電化の動作を説明したが、次に、本発明の次の特徴である、プログラム固有の省電化を可能にする技術について、次の図19を用いて説明する。尚、図3における説明と同様に、ここでは、プログラムの実行をプロセスの開始、プログラムの終了をプロセスの終了と呼び、既に開始されたプロセス内から、さらに別のプロセスを開始させるシステムについて説明する。

【0072】図19は、図1および図3における省電制御システムの本発明に係わる処理動作の一実施例を示すフローチャートである。本実施例は、プログラム固有の省電化を可能にするものであり、図3のプログラム26が、図3の実行プロセス制御部23に、新規プロセスの開始を要求すると、図3のアイドル状態のコンソール入

力センサ時間密度記憶部14内の現プロセスのアイドル状態のコンソール入力センサ回数の時間密度テーブル

（`dens_tbl`）を温存して、次のプロセスを起動するための準備を行なう。すなわち、まず、現プロセスを破棄するか否かを確認し（ステップ1001）、破棄する場合には、現在のプログラム終了アドレスを変更し（ステップ1002）、後述の図20で示すようにして、実行プロセス制御の終了動作を行なう（ステップ1003）。また、現プロセスを破棄しない場合には、現在実行中のプロセス番号（`process`）に基づき、アイドル状態のコンソール入力センサ回数の時間密度テーブル（`dens_tbl`）に空きがあるか否かを確認する（図中、`process?`）（ステップ1004）。もし、「`process<PROCESS_MAX`（温存できるプロセスの最大数）-1」で、空きがあれば、プロセス番号（`process`）をインクリメントする（ステップ1005）。そして、後述の図21に示す現プロセスの省電制御部の初期化を行なう（ステップ1006）。

【0073】このようにして、次のプロセスを起動するための準備が終了すれば、当該するプログラムのロードと、アイドル状態のコンソール入力センサ回数の時間密度テーブル（`dens_tbl`）の更新を行なう。すなわち、図3のプログラムロード／セーブ制御部13により、該当プログラムをロードし（ステップ1007）、そして、後述の図21で示す、この該当プログラム用の省電制御情報をロードする（ステップ1008）。そして、この省電制御情報で、図3のアイドル状態のコンソール入力センサ時間密度記憶部14内の、現プロセスのアイドル状態のコンソール入力センサ回数の時間密度テーブル（`dens_tbl`）を更新する（ステップ1009、1010）。

【0074】このようにして、当該するプログラムのロードと、アイドル状態のコンソール入力センサ回数の時間密度テーブル（`dens_tbl`）の更新が終了すれば、呼び出し元にプログラム終了アドレスを設定して（ステップ1011）、プログラムエントリへジャンプし（ステップ1012）、プロセスの実行を開始する。以上のように、プロセスの開始、終了を行なっても、図1におけるタイマ割込み制御部10で行なわれるアイドル状態の検出を毎回やり直すことなく、プロセス開始時から、以前に、そのプロセスが終了したときの省電制御を、再び開始できるので、無駄がない。

【0075】次に、このプロセスの終了について、次の図20を用いて説明する。図20は、図19における実行プロセスの終了制御動作の一実施例を示すフローチャートである。図3のプログラム26が、図3の実行プロセス制御部23に、現プロセスの終了を要求すると、図3のアイドル状態のコンソール入力センサ時間密度記憶部14に記録されている現プロセスのアイドル状態のコ

10

20

30

40

50

ンソール入力センサ回数の時間密度テーブルを、現プロセスの元となるプログラム固有の記憶域に保存する（図中、「現プロセスのdens__tblをセーブ」）（ステップ1101）。尚、本実施例では、後述の図22のように、現プロセスのアイドル状態のコンソール入力センサ回数の時間密度テーブルを、該当プログラムファイル自身に埋め込んでいる。

【0076】次に、図3の現実行プロセス記憶部18に前プロセスがあるか否かを確認し（図中、process?）（ステップ1102）、あれば、プロセス番号をディクリメントして（図中、--process）（ステップ1103）復帰し、呼び出し元にジャンプする（ステップ1104）。尚、最初の現プロセスの終了時には、ステップ1102においては、前プロセスなし（図中、==0）であり、次の図21で示す現プロセスの省電制御部初期化の処理を行なう（ステップ1105）。

【0077】図21は、図19および図20における現プロセスの省電制御部初期化の処理手順の一実施例を示すフローチャートである。前述の図15で示す処理により、現プロセスのアイドル状態のコンソール入力センサ回数の時間密度テーブル（dens__tbl）から、アドレスを抽出し（図中、addr=現プロセスのdens__tblアドレス）（ステップ1201）、それぞれの変数に初期値を設定する（ステップ1202）。すなわち、現在のアイドル状態のコンソール入力センサ回数の時間密度テーブル（dens__tbl）のアドレス（図中、addr[0]=0）、図1のコンソール入力センサ制御部24用のコンソール入力センサ回数（図中、sens__cnt=0）とコンソール入力センサ回数の時間密度測定用のタイマカウンタ（図中、sens__tmr=SENS__TMR__MAX）、および、図1のタイマ割込み制御部10用のコンソール入力センサ回数（図中、dens__cnt=0）とコンソール入力センサ回数の時間密度測定用のタイマカウンタ（図中、dens__tmr=SENS__TMR__MAX）、そして、コンソール入力センサ時間密度の頻度テーブル（図中、freq__tbl[0~FREQ__TBL__MAX-1]=0）の初期値の設定を行なう。

【0078】図22は、図19における該当プログラムの省電制御情報のロード処理で格納される省電制御情報の一実施例を示す説明図である。プログラムのコードやデータ部221に、プログラム実行制御情報222を付与して、実行可能プログラムファイル223が構成されている。また、プログラム実行制御情報222は、この省電制御情報が有効となるBIOSや、ハードウェアバージョン識別子、および、当該するアイドル状態のコンソール入力センサ回数の時間密度テーブル（dens__tbl）のコピーからなる。このような構成のプログラム実行制御情報222を、該当するプログラムに付与し

て格納することにより、既に、一度でも、アイドル状態が検出されたプログラムであれば、二度目からは、すぐに、省電効果が期待できる。

【0079】次に、本発明のもう一つの特徴である、省電制御条件をプログラム毎に変更する処理に関して、次の図23を用いて説明する。図23は、図4における省電制御システムの本発明に係わる省電制御条件の設定制御の一実施例を示すフローチャートである。図4で示す省電制御システムの省電制御は、例えば、図10~図22のような実施例のように実現されるが、この中には、省電制御の条件を表現するものがいくつかある。これらを変更することにより、アイドル検出の具合や、省電化する装置の種類などを変更できる。すなわち、省電制御条件の設定要求があれば（ステップ1301）、コンソール入力センサ回数（dens__cnt、sens__cnt）を、コンソール入力センサ時間密度とみなすまでに発生するタイマ割込みの数（図中、SENS__TMR__MAX）から、省電動作モードの各フェーズを設定するための制御テーブルのアドレス（図中、sleep__tblアドレス）までの、それぞれの省電制御条件を、指定のバッファ内の値で変換し（ステップ1302）、その他の変数を、図21で示した現プロセスの省電制御部初期化により初期値に設定する（ステップ1303）。

【0080】ステップ1301において、新たな設定要求がなければ、現在の各変数の値を指定バッファに格納して（ステップ1304）、処理を終了する。この処理は、図4において、プログラム26が、省電制御条件の設定制御部27へ要求すれば良い。省電制御条件の設定制御部27は、条件記憶部28の変更を行なうが、条件記憶部28は、省電制御の各ブロックから参照されているので、省電制御条件を一括して変更できる。

【0081】以上、図1~図23を用いて説明したように、本実施例の省電制御システムでは、コンソール入力センサを行なう各々のプログラムに対しても、それぞれの処理に適応して、そのアイドル状態を確実に検出できるので、本発明に係わる省電化の制御を関知していないプログラムが動作していても、システムの消費電力の低減を効率良く行なうことができる。

【0082】また、コンソール入力センサを行なわないプログラムでは、直ちに省電動作モードとすることができるので、さらにシステムの消費電力を低減できる。また、アイドル状態のコンソール入力センサ時間密度は、プログラムによって、固有の値を持つことが考えられるが、プログラム内の処理部（コード部）は、何度実行されても、通常は、変わることがなく、一度検出したアイドル状態のコンソール入力センサ時間密度を、そのプログラム固有の領域に保存しておき、再実行時に、これを利用すれば、実行直後から、省電動作モードにすることができ、さらに、消費電力を低減できる。

【0083】また、通常、装置を省電動作モードにすると、次に、通常動作モードへ移行するのに、ある程度の時間を必要とする場合が多く、また、この復帰時間は、装置の種類によってまちまちであるが、本実施例では、装置別に、省電動作モードの設定手段を設け、アイドル状態検出後、この状態が続く限り、省電動作モードから、通常動作モードへの復帰時間が短い順序で、段階的に、省電動作モードに設定でき、スループットの損失を抑えた効率の良い消費電力制御を行なうことができる。

【0084】また、あらゆるプログラムで、省電化の効果が最大になるように最適化された各種の条件のもとに省電制御が行なわれるが、このことは、逆に、あるプログラムでは、未だ、省電化の余地が残されているかもしれないことを意味する。本実施例では、この各種条件を、プログラム個別に最適化されたものにすることができるので、さらに、システムの消費電力を低減できる。尚、例えば、図 1 において、集中的に発生したコンソール入力センサ回数の時間密度の抽出で述べたと同様に、本発明は、図 1 ～ 図 2 3 を用いて説明した実施例に限定されるものではない。

【0085】

【発明の効果】本発明によれば、標準入力装置に対する入力センサを行なう各々のプログラムに対して、CPU や表示装置など常時使用されているものを含み、それぞれの装置を、復帰時間の短い装置から、任意に設定された時間間隔で、段階的に省電動作モードに設定して行くことができ、コンピュータを利用した機器の省電化を効率良く行なうことが可能である。

【0086】

【図面の簡単な説明】

【図 1】本発明を施したコンピュータ利用機器の省電制御システムの本発明に係わる構成の第 1 の実施例を示すブロック図である。

【図 2】本発明を施したコンピュータ利用機器の省電制御システムの本発明に係わる構成の第 2 の実施例を示すブロック図である。

【図 3】本発明を施したコンピュータ利用機器の省電制御システムの本発明に係わる構成の第 3 の実施例を示すブロック図である。

【図 4】本発明を施したコンピュータ利用機器の省電制御システムの本発明に係わる構成の第 4 の実施例を示すブロック図である。

【図 5】本発明のコンピュータ利用機器の省電制御システムの本発明に係わるアイドル状態の処理動作の一実施例を示すフローチャートである。

【図 6】図 1 0 ～ 図 2 3 の説明で用いられる変数の内容を示す説明図である。

【図 7】図 1 0 ～ 図 2 3 の説明で用いられる条件定数の内容を示す説明図である。

【図 8】図 6 における各変数のデータ構造を示す説明図

その 1 である。

【図 9】図 6 における各変数のデータ構造を示す説明図その 2 である。

【図 1 0】図 1 におけるタイマー割込み制御部の本発明に係わる処理動作の一実施例を示すフローチャートである。

【図 1 1】図 1 におけるタイマー割込み制御部の頻度テーブルの解析とアイドル状態のコンソール入力センサ時間密度の抽出および登録処理動作の一実施例を示すフローチャートである。

【図 1 2】図 1 における省電制御システムのコンソール入力装置からの割込みに基づく処理動作の一実施例を示すフローチャートである。

【図 1 3】図 1 における省電制御システムの印刷装置および外部記憶装置からの割込みに基づく処理動作の一実施例を示すフローチャートである。

【図 1 4】図 1 におけるコンソール入力センサ制御部の本発明に係わる処理動作の一実施例を示すフローチャートである。

【図 1 5】図 3 の省電制御システムにおける実行プロセス制御部の本発明に係わる処理動作の一実施例を示すフローチャートである。

【図 1 6】図 1 および図 3 における省電制御システムのプログラムからの周辺装置の要求に基づく処理動作の一実施例を示すフローチャートである。

【図 1 7】図 1 および図 4 における省電制御システムの本発明に係わる省電動作モードの設定制御手順の一実施例を示すフローチャートである。

【図 1 8】図 2 の省電制御システムにおけるコンソール入力制御部の本発明に係わる処理動作の一実施例を示すフローチャートである。

【図 1 9】図 1 および図 3 における省電制御システムの本発明に係わる処理動作の一実施例を示すフローチャートである。

【図 2 0】図 1 9 における実行プロセスの終了制御動作の一実施例を示すフローチャートである。

【図 2 1】図 1 9 および図 2 0 における現プロセスの省電制御部初期化の処理手順の一実施例を示すフローチャートである。

【図 2 2】図 1 9 における該当プログラムの省電制御情報のロード処理で格納される省電制御情報の一実施例を示す説明図である。

【図 2 3】図 4 における省電制御システムの本発明に係わる省電制御条件の設定制御の一実施例を示すフローチャートである。

【符号の説明】

- 1 表示装置
- 2 印刷装置
- 3 外部記憶装置
- 4 コンソール入力センサ時間密度の発生頻度記憶部

29

30

5 タイマ発振装置

6 コンソール入力装置

7 割込み制御部

8 表示・印刷制御部

9 外部記憶制御部

10 タイマ割込み制御部

11 コンソール入力割込み制御部

12 通常動作モード設定制御部

13 プログラムロード/セーブ制御部

14 アイドル状態のコンソール入力センス時間密度記憶部

15 時間計測用タイマカウンタ記憶部

16 コンソール入力データ記憶部

17 現動作モード記憶部

18 現実行プロセス記憶部

* 19 コンソール入力センス回数記憶部

20 コンソール入力データ変換部

21 省電動作モード設定制御部

22 周辺入出力サービス制御部

23 実行プロセス制御部

24 コンソール入力センス制御部

25 コンソール入力制御部

26、26a プログラム

27 省電制御条件の設定制御部

28 条件記憶部

29 省電動作モードフェーズ制御記憶部

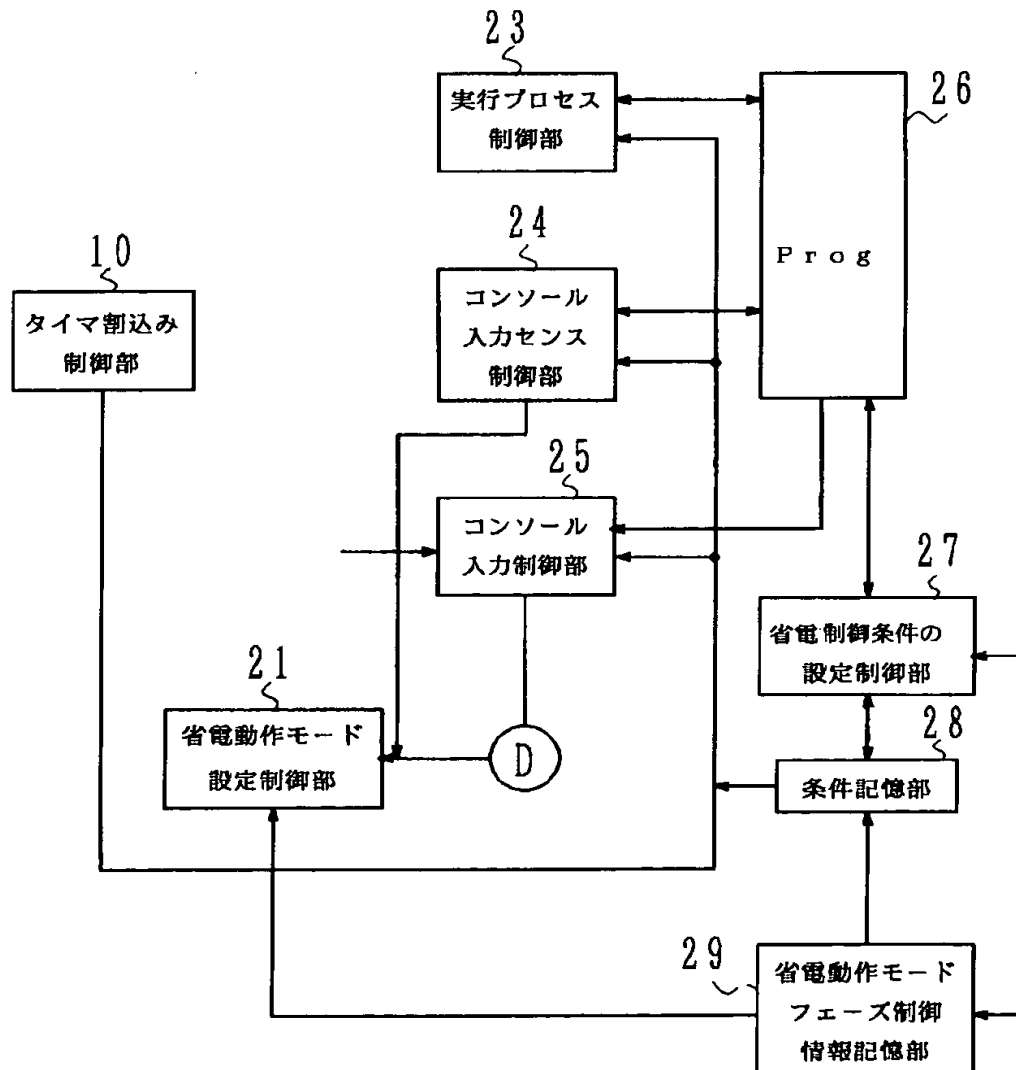
221 プログラムのコード、データ部

222 プログラム実行制御情報

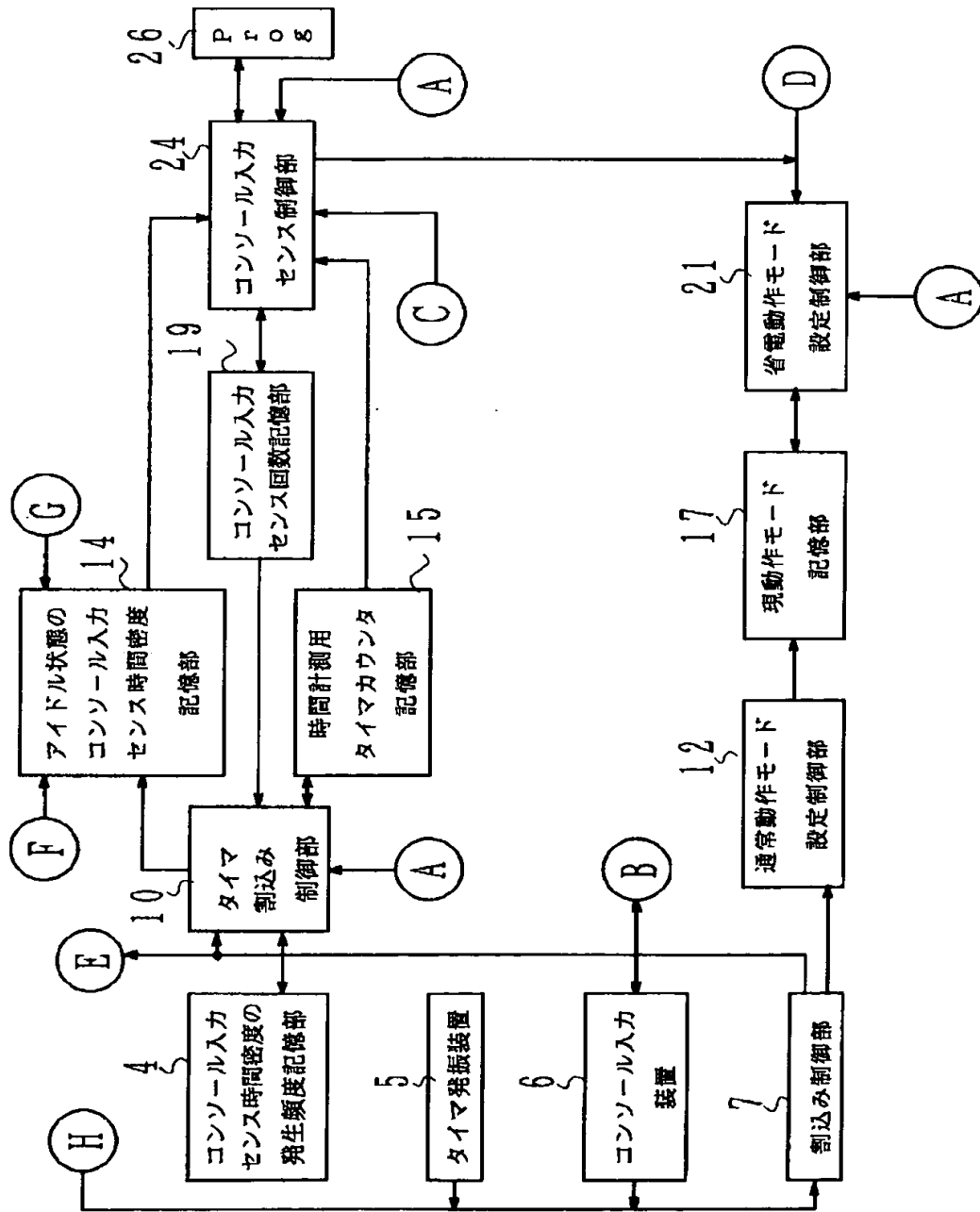
223 実行可能プログラムファイル

* L1、L2 ループ

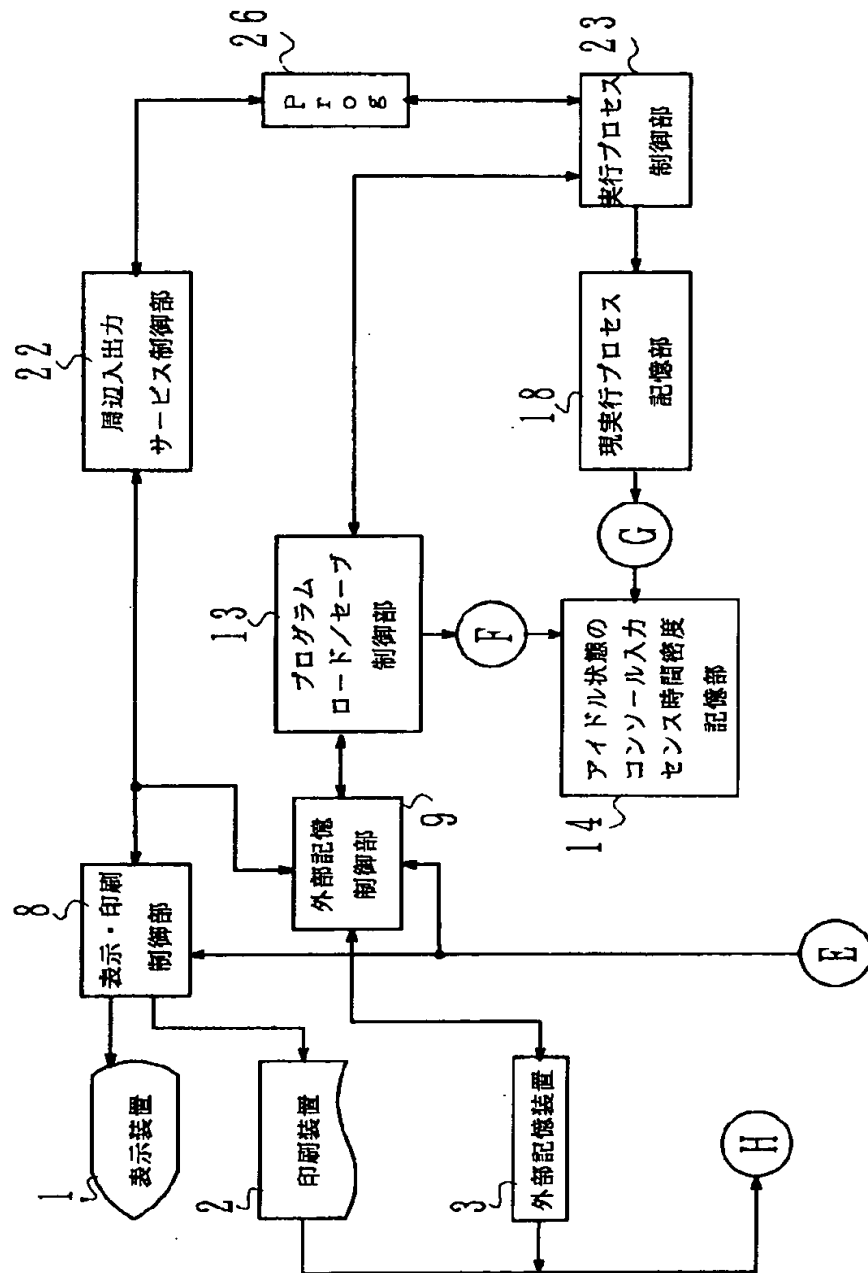
【図4】



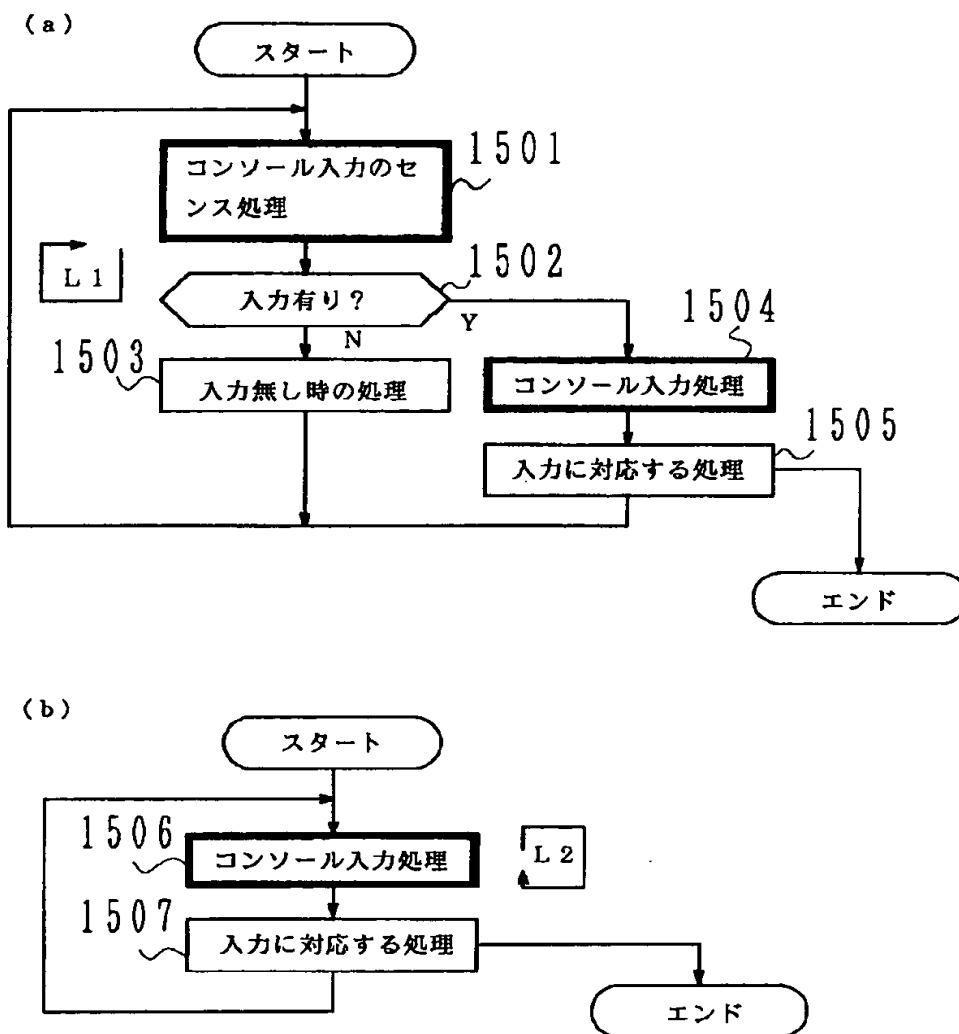
【図 1】



【図 3】

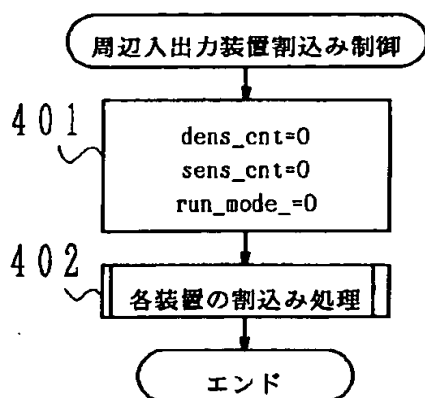


【図5】



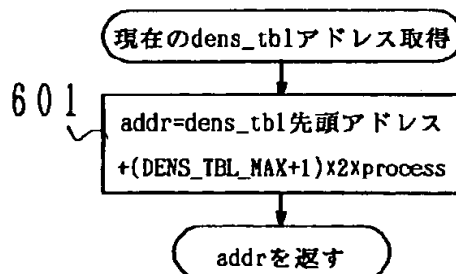
【図13】

周辺入出力装置割込み制御のフローチャート



【図15】

現在dens_tblアドレス取得のフローチャート



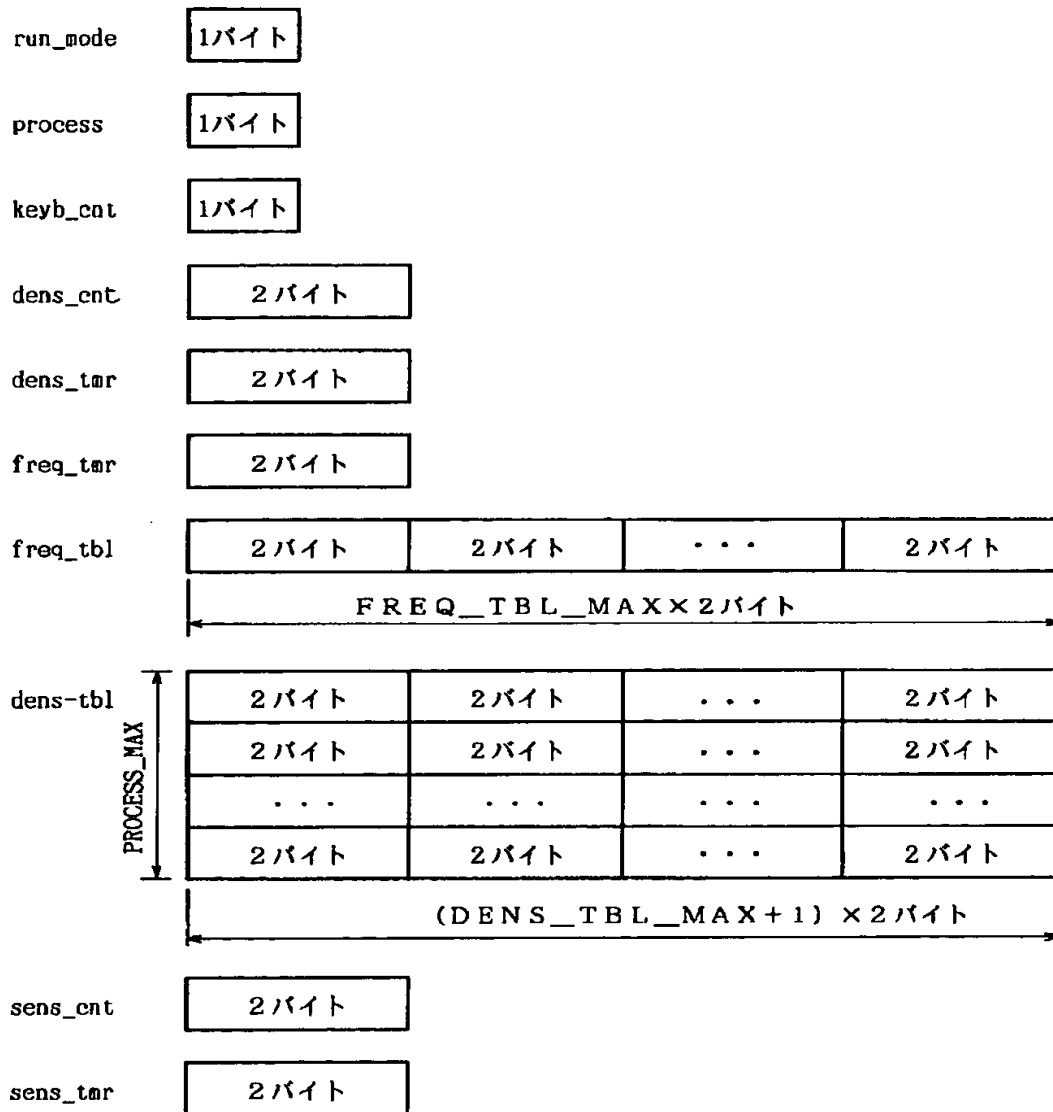
フローチャートで使用される変数

変数名	内容	初期値	各ブロック図 内符号
run_mode	現在の動作モード (0: 通常動作モード, 0以外: 省電動作モード)	0	17
process	現在実行中のプロセス番号 (0~PROCESS_MAX-1)	0	18
keyb_cnt	現在までに入力されている有効データ数 (0~記憶バッファの大きさまで)	0	16
dens_cnt	コンソール入力センサ回数	0	19
dens_tmr	コンソール入力 センサ時間密度測定用タイマカウンタ	SENS_TMR_MAX	15
freq_tmr	コンソール入力センサ時間密度の頻度測定用タイマカウンタ	FREQ_TMR_MAX	15
freq_tbl	コンソール入力センサ時間密度の頻度テーブル	全0	4
dens_tbl	アイドル状態のコンソール入力センサ時間密度テーブル	全0	14
sens_cnt	コンソール入力センサ回数	0	19
sens_tmr	コンソール入力センサ時間密度測定用タイマカウンタ	SENS_TMR_MAX	15
sleep_tmr	省電動作モードの各フェーズを設定するためのタイマカウンタ	run_modelによる	15
sleep_tbl	省電動作モードの各フェーズを設定するための制御テーブル	図9参照	29
active_tmr	省電動作モードから通常動作モードへ復帰時の待ち時間	run_modelによる	15

【図6】

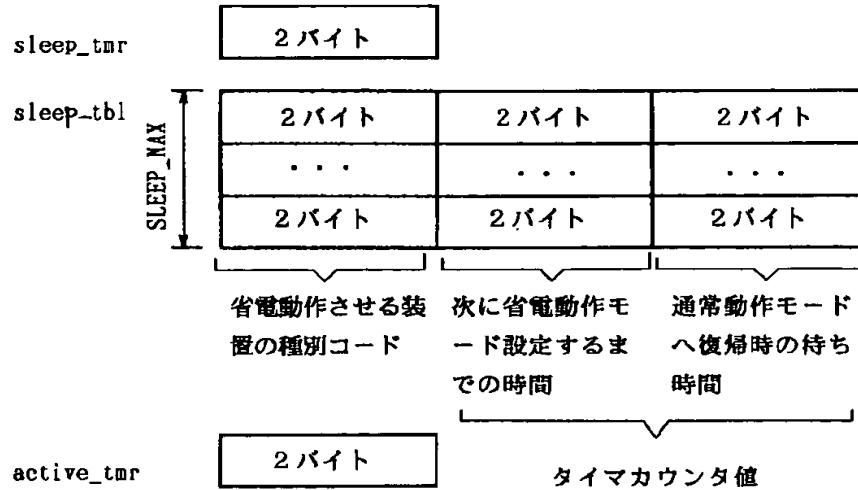
【図8】

フローチャートで使用する変数データ構造 (その1)

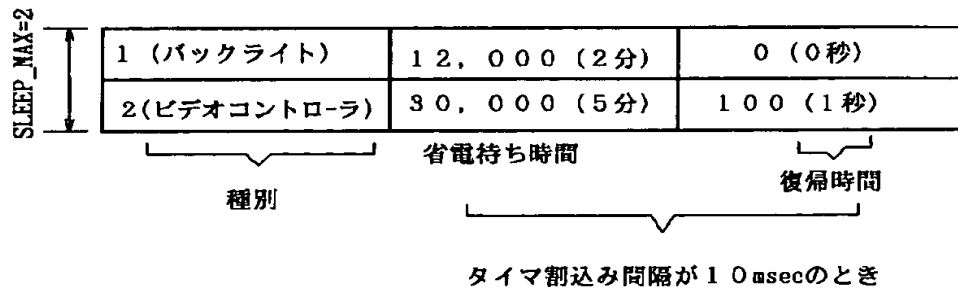


【図9】

(a) フローチャートで使用される変数データ構造 (その2)

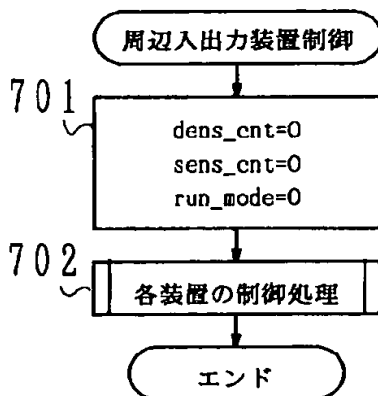


(b) sleep_tbl の例



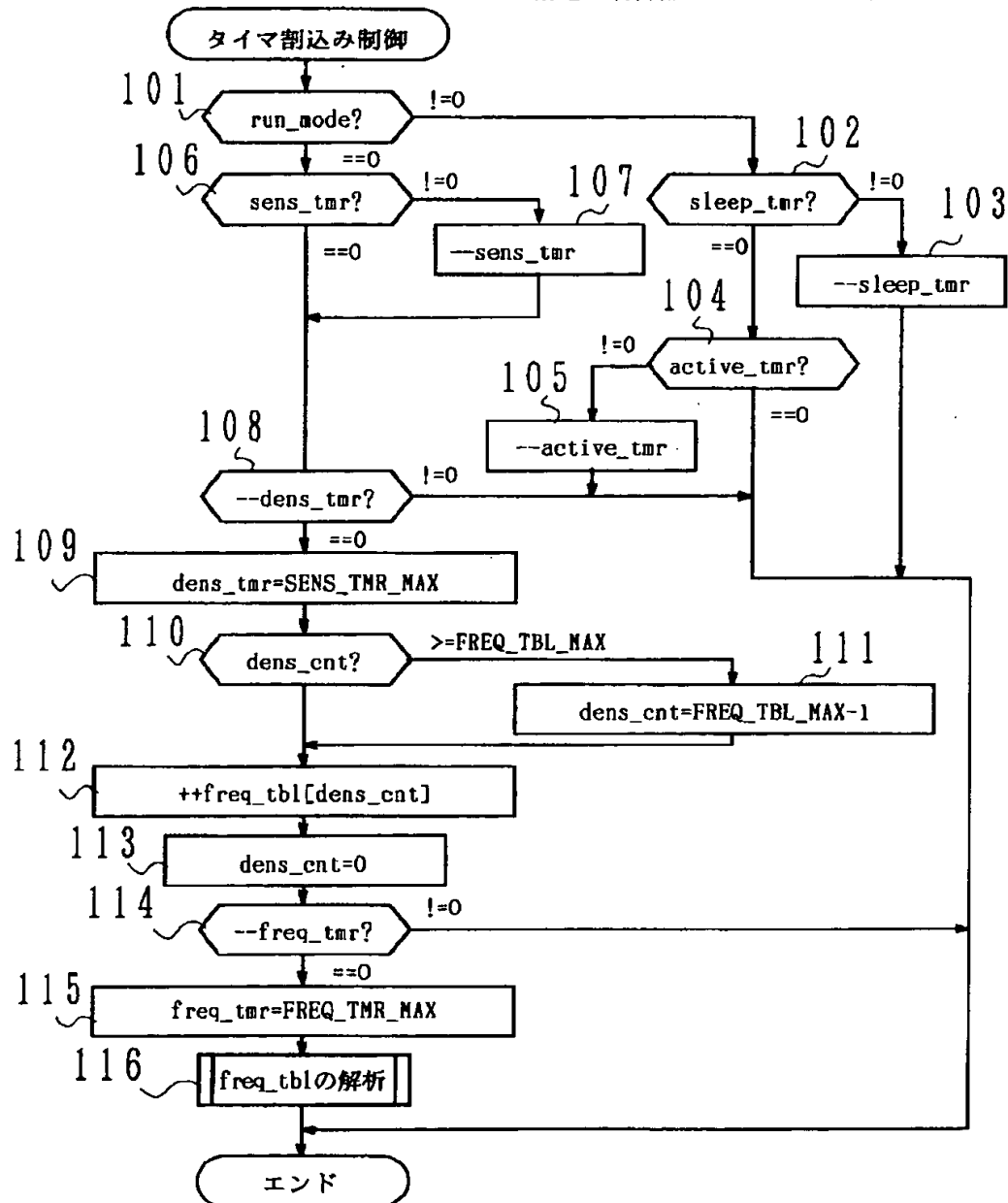
【図16】

周辺入出力装置制御のフローチャート



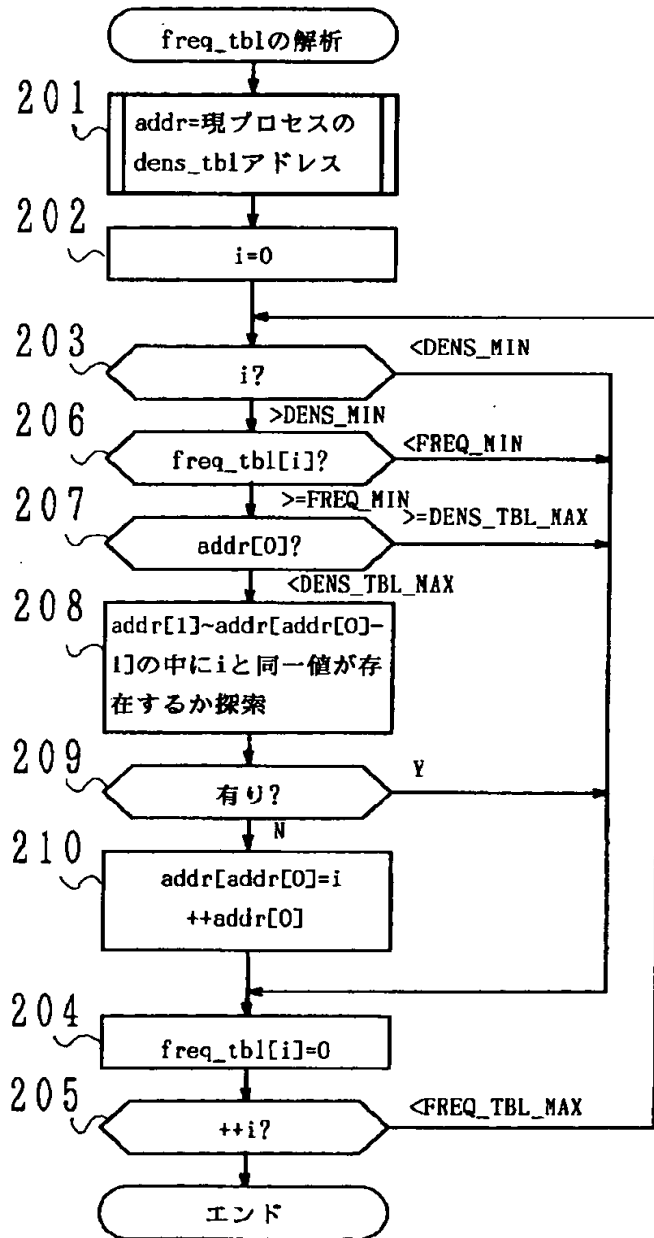
【図10】

タイマ割込み制御部のフローチャート



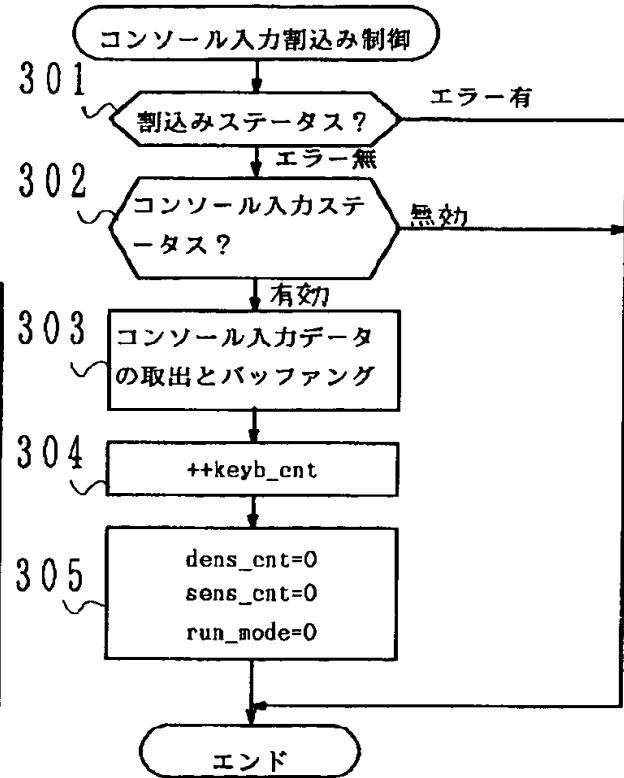
【図11】

freq_tbl解析のフローチャート



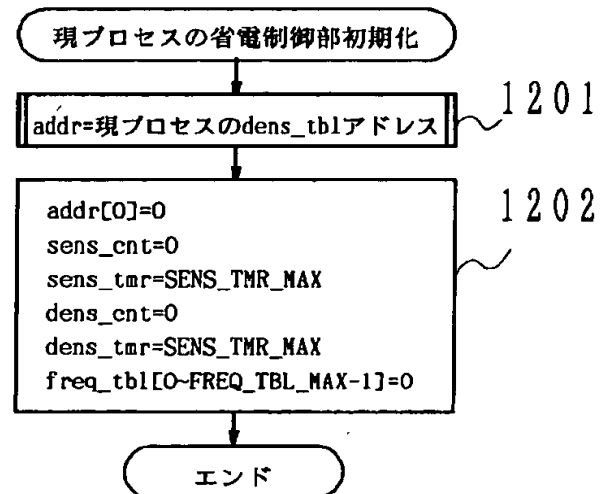
【図12】

コンソール入力割込み制御のフローチャート



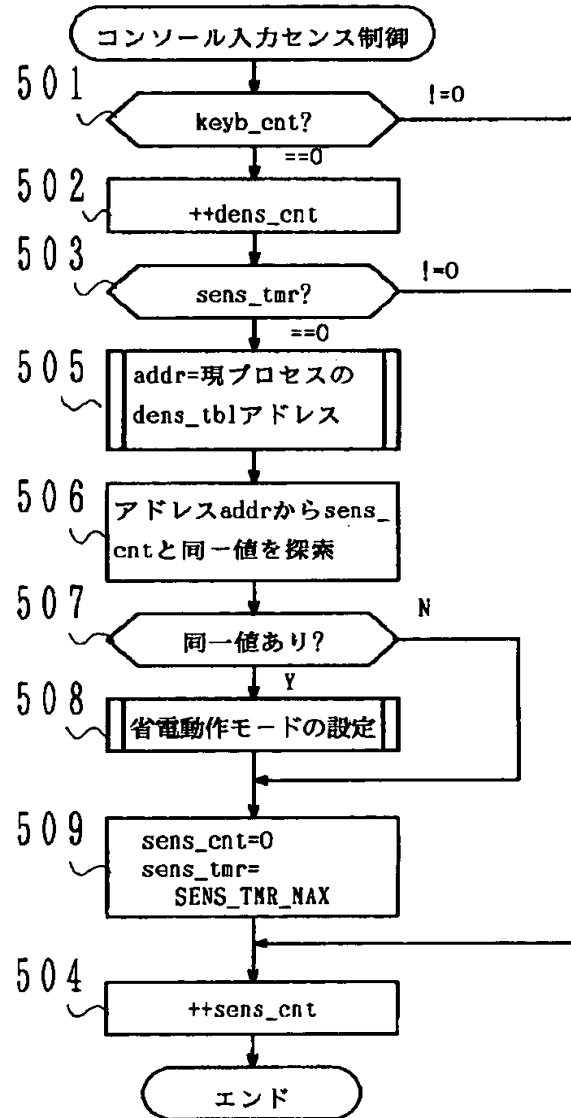
【図21】

現プロセスの省電制御部初期化のフローチャート



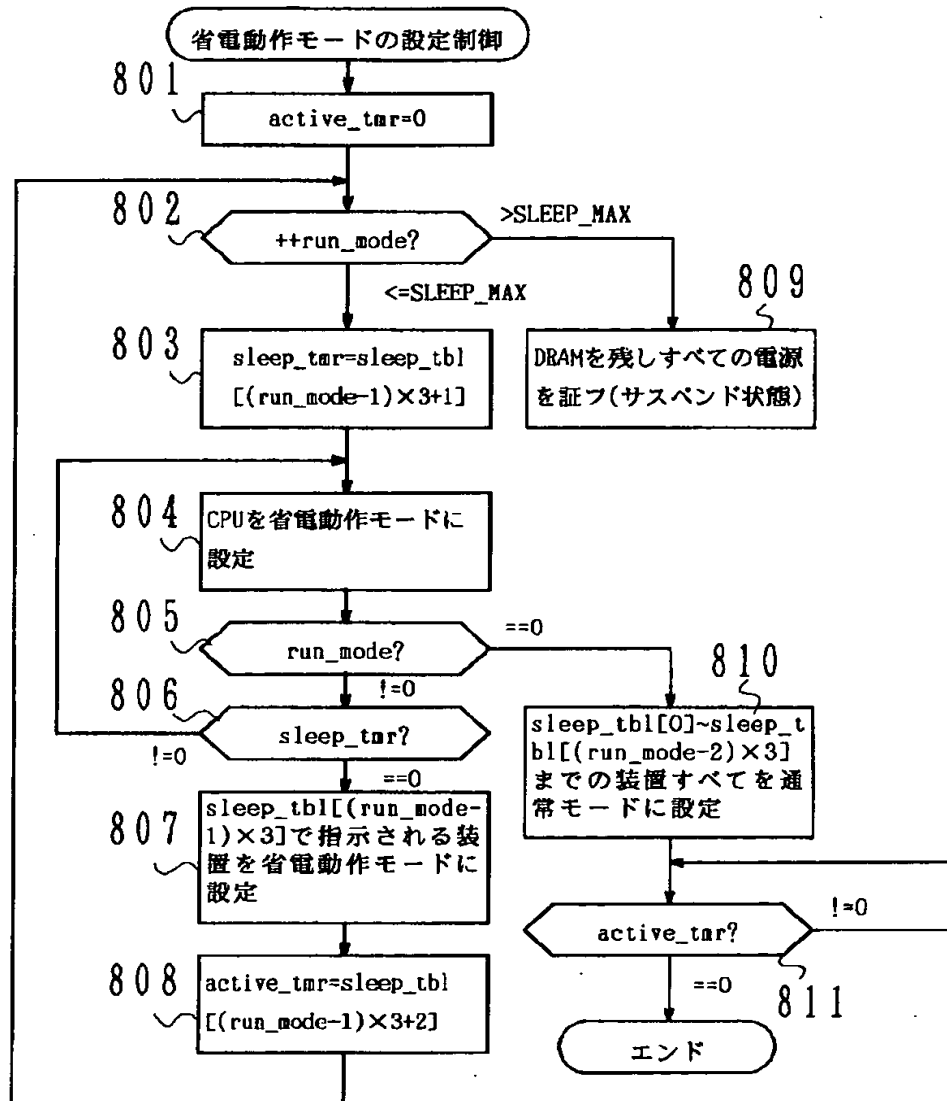
【図14】

コンソール入力センス制御のフローチャート



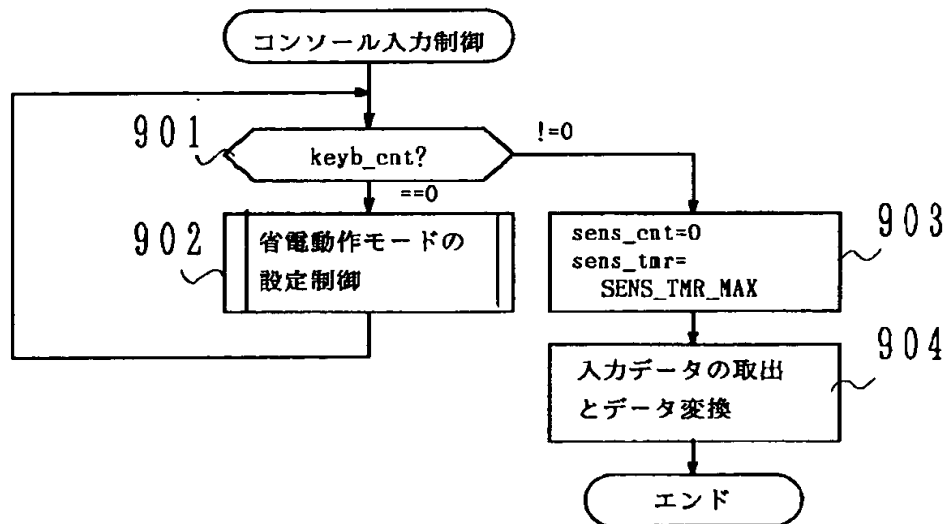
【図17】

省電動作モードの設定制御のフロー



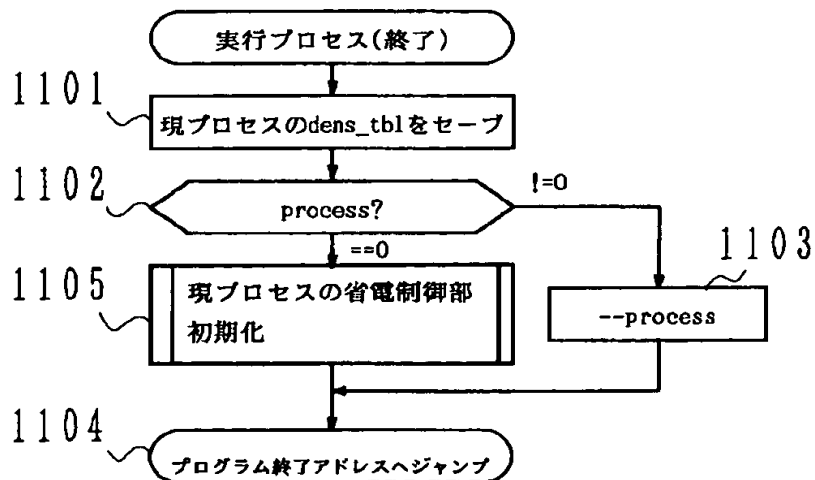
【図18】

コンソール入力制御のフローチャート



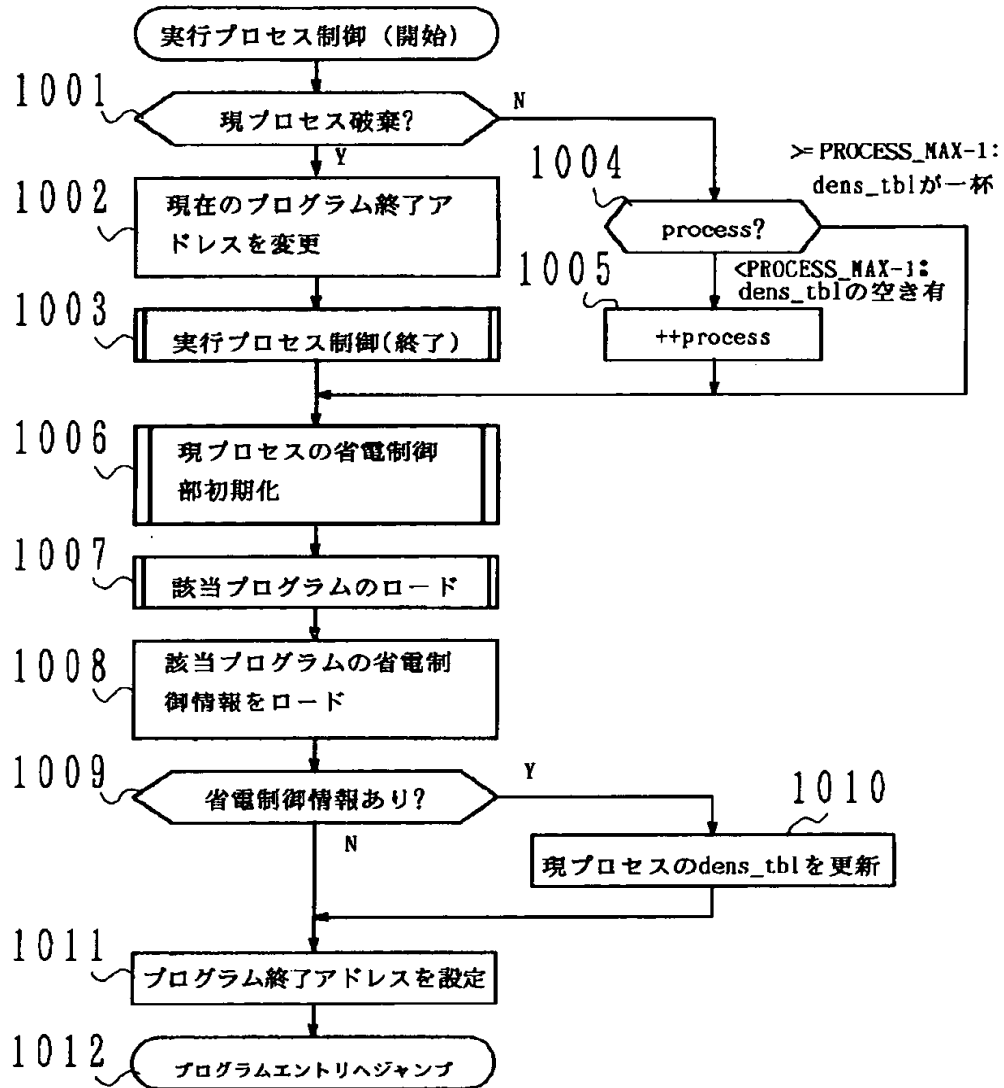
【図20】

実行プロセス(終了)のフローチャート



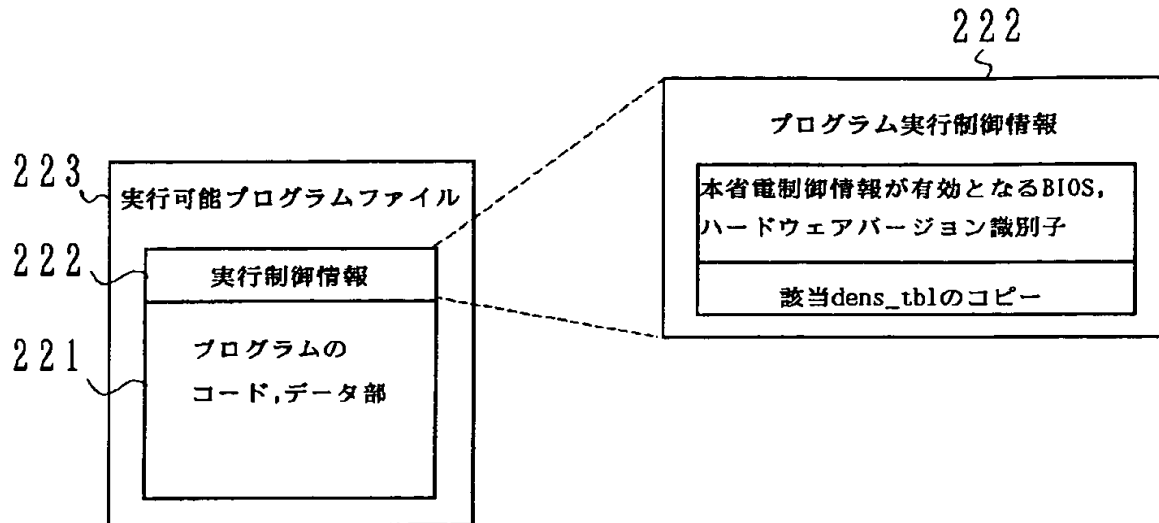
【図19】

実行プロセス制御（開始）のフローチャート



【図22】

プログラムの省電制御情報の格納例



【図23】

省電制御条件の設定制御のフローチャート

